An Order-Based Theory of Updates for Closed Database Views †

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

Minor corrections: 05 August 2006

Abstract

The fundamental problem in the design of update strategies for views of database schemata is that of selecting how the view update is to be reflected back to the base schema. This work presents a solution to this problem, based upon the dual philosophies of closed update strategies and order-based database mappings. A closed update strategy is one in which the entire set of updates exhibit natural closure properties, including transitivity and reversibility. The order-based paradigm is a natural one; most database formalisms endow the database states with a natural order structure, under which update by insertion is an increasing operation, and update by deletion is decreasing. Upon augmenting the original constant-complement strategy of Bancilhon and Spyratos – which is an early version of a closed update strategy – with compatible order-based notions, the reflection to the base schema of any update to the view schema which is an insertion, a deletion, or a modification which is realizable as a sequence of insertions and deletions is shown to be unique and independent of the choice of complement.

In addition to this uniqueness characterization, the paper also develops a theory which identifies conditions under which a natural, maximal, update strategy exists for a view. This theory is then applied to a ubiquitous example – single-relational schemata constrained by equality-generating dependencies. Within this framework it is shown that for a view defined as a projection of the main relation, the only possibility is that the complement defining the update process is also a projection, and that the reconstruction is based upon functional dependencies.

†A preliminary version of parts of this paper appeared as reference [16].

1. Introduction

Two essential features of any large, modern database management system are the capability of performing updates and the capability of delivering limited access through views. The integration of these two aspects — the capability of supporting updates which are specified in views — has long been known to be a very difficult problem. The major problem is not one of existence; it is always possible to reflect an update to a view schema back to the main schema. Rather, the key issue is one of suitability; the reflection of the view update back into the base schema must be acceptable according to certain criteria. The manner in which such suitability criteria are determined is one for which a variety of philosophies have emerged.

1.1 Open vs. closed update strategies In [12], the extremes of suitability criteria for view updates have been termed open and closed update strategies. Roughly speaking, an *open* strategy is very liberal; as many updates as possible are allowed, with the user expected to be aware of the consequences of such a strategy. A *closed* strategy, on the other hand, is very conservative and systematic. Roughly speaking, there are two principles which govern a closed strategy. First of all, the admissibility of an update to the view must depend only upon the state of the view; it must never necessary to have further information about the state of the base schema in order to determine whether an update is to be allowed. Second, the family of allowed updates must form an equivalence relation; that is, it must be closed under reversibility, transitivity, and of course identity. In short, a closed view appears as a schema unto itself, and the family of updates which is allowed looks exactly as would a family of updates to a base schema. These ideas are elaborated more fully in 1.2 below.

As the focus of this paper is support for updates under closed strategies, it is fair to ask why one should consider such strategies at all, given that open strategies invariably allow a wider range of updates to the view. The short answer is that open strategies inevitably give rise to certain anomalies which render them less than suitable in certain situations. A few examples will illustrate this point. Let $\mathbf{C}_0$ denote the schema with the single relation symbol $P[\mathsf{Name}, \mathsf{Dept}, \mathsf{Proj}]$, and the initial instance $M_0 = \{(\text{Smith}, 1, \text{A}), (\text{Jones}, 2, \text{A}), (\text{Jones}, 2, \text{B})\}$. The informal semantics of a tuple such as $(\text{Smith}, 1, \text{A})$ is that employee Smith works in department 1 and on project A. It is furthermore assumed that the functional dependency $\mathsf{Name} \rightarrow \mathsf{Dept}$ holds; *i.e.*, that an individual works in only one department. On the other hand, an individual may work on several projects.

First, let $\Pi_{\mathsf{NP}} = (P[\mathsf{Name}, \mathsf{Proj}], \pi_{(\mathsf{Name}, \mathsf{Proj})})$ denote the view which retains just the Name and Proj attributes; the image of M_0 under this view is $\{(\text{Smith}, \text{A}), (\text{Jones}, \text{A}), (\text{Jones}, \text{B})\}$. Consider deletion of the tuple (Smith, A). Under an open strategy, it might be argued that such an operation should be allowed, as the only reasonable reflection is to delete $(\text{Smith}, 1, \text{A})$ from M_0. While this reflection is unique and unambiguous, there are nonetheless at least two points of caution. First of all, this update involves a *hidden trigger*, in that information about the department in which Smith works is also deleted. The user of the view must have access to information beyond the view Π_{NP} itself in order to be aware of this. Furthermore, this update is *irreversible*, in the sense that it is not possible to undo its effect by re-inserting (Smith, A) into the view, since the information regarding the department of Smith has been lost. For these reasons, this update would not be allowed in a closed view. On the other hand, these difficulties do not arise in the deletion of (Jones, A); in this case the view update may be realized via deletion of (Jones, 2, A) in M_0. Since information about the department of Jones is retained in the tuple (Jones, 2, B), and since $\mathsf{Name} \rightarrow \mathsf{Dept}$ holds, re-insertion of (Jones, A) is unambiguous and restores the initial state M_0. Thus, deletion of (Jones, A) would be allowed, even in a closed strategy, from the view state resulting from M_0.

It might be argued that this problem would be avoided were the original schema normalized via decomposition. Indeed, if the original schema were decomposed losslessly into Π_{NP} and $\Pi_{\mathsf{ND}} = (P[\mathsf{Name}, \mathsf{Dept}], \pi_{(\mathsf{Name},\mathsf{Dept})})$, with the latter schema embodying the functional dependency $\mathsf{Name} \rightarrow \mathsf{Dept}$, then the problems described above would disappear, since Π_{NP} would be completely independent of Π_{ND} and so could be updated without restriction. However, this new schema is too liberal in its constraints; there is no longer any connection between the employees in the the two component relations.

A more plausible decomposition would enforce the foreign-key constraint $\pi_{\mathsf{Name}}(\Pi_{\mathsf{NP}}) \subseteq \pi_{\mathsf{Name}}(\Pi_{\mathsf{ND}})$; that is, every employee who is associated with a project must also be associated with a department. Call the resulting two-relation schema $\mathbf{C}_1$. In comparison with the previous example, many of the complications surrounding deletion of the tuple (Smith, A) disappear. Indeed, this tuple may be deleted from the state of Π_{NP} without altering the state of Π_{ND} and without violating the foreign-key constraint, and so it may subsequently be re-inserted without problem. In other words, this update is reversible, and involves no hidden triggers. However, there remains a difficulty; namely, the admissibility of an insertion into the state of the view depends upon information not visible within that view. More specifically, suppose that the state of Π_{ND} is now $M_{1_{ND}} = \{(\text{Smith}, 1), (\text{Jones}, 2), (\text{Wilson}, 1)\}$, with the state of Π_{NP} now $M_{1_{NP}} = \{(\text{Smith}, \text{A}), (\text{Jones}, \text{A}), (\text{Jones}, \text{B})\}$. The state of the full schema is thus the pair $(M_{1_{ND}}, M_{1_{NP}})$. It is clear that the tuple (Wilson, C) may be inserted into the state of the view Π_{NP}; on the other hand, the tuple (Young, C) may not be so inserted, since no department information is available for Young. This situation involves a *hidden dynamic constraint*, in the sense that whether or not the insertion is permitted depends upon information not contained in the view itself. For this reason, under a closed interpretation, insertion of (Wilson, C) would be disallowed, although it might be allowed under some open strategies.

The above example $\mathbf{C}_1$ is particularly interesting because it has been normalized to Boyce-Codd Normal Form (BCNF) [1, 11.2.1]. Thus, the process of normalization to BCNF does not guarantee that a view consisting of one of the component relations will be closed. Of course, this example includes a foreign-key dependency, but it is a natural component of the normalization process, and such dependencies are supported by standard SQL and virtually all modern relational database systems, and would be enforced in a real application.

In order to allow updates to a view corresponding to Π_{NP} within a closed framework, it is necessary to modify the schema slightly. One possibility is to allow null values in the Proj field of Π_{NP}, and to enforce the full domain-equality constraint $\pi_{\mathsf{Name}}(\Pi_{\mathsf{ND}}) = \pi_{\mathsf{Name}}(\Pi_{\mathsf{NP}})$. Call the resulting two-relation schema $\mathbf{C}_2$, and let the initial states of Π_{ND}

and Π_{NP} be $M_{2_{ND}} = M_{1_{ND}}$ and $M_{2_{NP}} = M_{1_{NP}} \cup \{(\text{Wilson}, \text{NULL})\}$, respectively. Here NULL represents a "not exists" null, in the sense that Young is not associated with any project. Now the semantic equivalent of the insertion of (Wilson, C) into the state of Π_{NP} is allowed; the tuple (Wilson, NULL) is simply replaced with this new value. Insertion of (Young, C) is still not allowed; however, the information necessary to make this distinction is now embodied in the view. Deletions of tuples not containing nulls are allowed, subject to the constraint that the deletion of the last tuple containing the name of an employee must be accompanied by the insertion of a tuple containing that name and a null in the project field. Deletions of tuples containing nulls is not allowed.

It is not the purpose of this paper to argue that closed update strategies are better than open ones, or that one should be used to the exclusion of the other. Indeed, no such argument seems possible, as they each have their place. Open strategies are useful as tools for experienced users who have wide access to the database, but wish, for convenience, to represent certain operations within a view. Such users can and will understand the more global implications of view updates, and so issues such as hidden triggers, irreversibility, and hidden dynamic constraints pose no real difficulties. On the other hand, there is also a clear need for views which are totally encapsulated. Many users, for reasons of security, lack of expertise, or simply effective management of complex schemata, need to work with views which, for operational purposes, are self contained. In particular, in such situations, the admissibility and effects of updates must be understandable within, and restricted entirely to, the view itself. In this context, anomalies such as hidden triggers, hidden dynamic constraints, and irreversibility must be avoided. It is to such closed update strategies which this paper is directed.

1.2 A formalization of closed update strategies The preceding discussion provides only an anecdotal characterization of the notion of a closed view; it remains to formalize this concept. Fortunately, to a large degree, this has already been done. The seminal work on closed strategies is the *constant-complement* approach, developed more than twenty years ago by Bancilhon and Spyratos [3]. The idea is quite simple. To support updates to the view Γ of the main schema $\mathbf{D}$, a view Γ' which is complementary to Γ is identified; *i.e.*, $\{\Gamma, \Gamma'\}$ forms a lossless decomposition of $\mathbf{D}$. Then, the only updates to Γ which are allowed are those which hold the state of Γ' fixed. Intuitively, the changes are isolated in Γ; the "rest" of the schema, which is Γ', cannot change. This freezing of the state of the complement under view update eliminates the possibility of hidden triggers and hidden dynamic constraints. Remarkably, it also eliminates the possibility of anomalies such as irreversibility.

An example will help illustrate. Let the base schema $\mathbf{E}_1$ have the single relation $R[ABC]$, governed by the functional dependency $B \to C$. Let the view to be updated be $\Pi_{AB} = (R[AB], \pi_{AB})$, the projection of the main relation onto the attributes AB. (In the context of the example of 1.1, just take $A = \mathsf{Proj}$, $B = \mathsf{Name}$, and $C = \mathsf{Dept}$.) A natural complement to Π_{AB} is the view $\Pi_{BC} = (R[BC], \pi_{BC})$. It is easy to see that updates to Π_{AB} which keep Π_{BC} constant are precisely those which hold the projection $\Pi_B = (R[B], \pi_B)$ of the relation $R[AB]$ constant. The common view Π_B is called the *meet* of Π_{AB} and Π_{BC}, and the latter two views are called Π_B-complements. In [12, 2.10], it is established that

every constant-strategy in the sense of [3] is in fact based upon a meet complement, and the updates which are allowed are precisely those which hold that meet constant.
This situation is illustrated diagrammatically in Fig. 1 to the right. The area with up-to-right crosshatch is to be held constant; this includes in particular the meet $\Pi_B = \Pi_{AB} \wedge \Pi_{BC}$. The rest of Π_{AB}, with down-to-right crosshatching, may be updated without restriction.

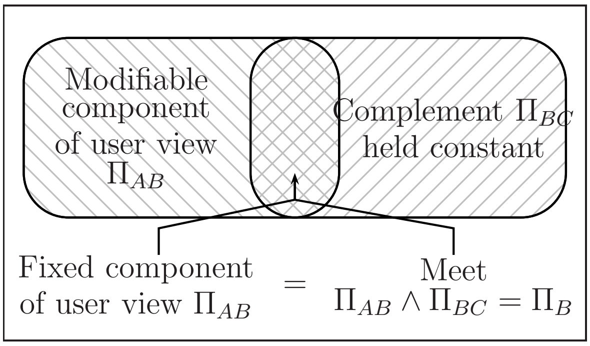


Figure 1: Visualization of the constant complement strategy

What is even more remarkable about the constant-complement approach is that it has an equivalent characterization in terms of admissible operations on the view, quite independent of any reference to the complement view Γ'. Specifically, let $\mathsf{LDB}(\mathbf{D})$ denote the states of the base schema $\mathbf{D}$, with $\mathsf{LDB}(\mathbf{V})$ denoting the states of the schema $\mathbf{V}$ of the view Γ. Write updates as ordered pairs. Thus, $(N_1, N_2) \in \mathsf{LDB}(\mathbf{V}) \times \mathsf{LDB}(\mathbf{V})$ represents an update from N_1 to N_2 in the view $\mathbf{V}$. The axioms of the constant complement strategy are expressed, somewhat informally, as follows.[1]

(cc:1) The admissibility of a view update (N_1, N_2) does not depend upon the state M_1 of the base schema which gives rise to N_1.

(cc:2) The identity update (N_1, N_1) to the view is always allowed, and is reflected as the corresponding identity in the base schema.

(cc:3) View updating is transitive, in the sense that if (N_1, N_2) and (N_2, N_3) are both allowed, then so too is (N_1, N_3), and the final state of the base schema reached after applying (N_1, N_2) and (N_2, N_3) in sequence is the same as that reached via (N_1, N_3) directly.

(cc:4) All view updates are reversible; if (N_1, N_2) is allowed, then so too is (N_2, N_1). It follows from (cc:2) and (cc:3) that the composition of these two must reflect to the identity update on the base schema.

It should be noted how natural these constraints are, in terms of providing a consistent and anomaly-free update interface. In particular, condition (cc:4) rules out hidden triggers and irreversible updates, while (cc:1) excludes hidden dynamic constraints.

1.3 The non-uniqueness of complements In many realistic examples, the choice of the complement view Γ' is "obvious." Nonetheless, from a purely mathematical point of view, there may be other possibilities. Consider the example of a relational schema $\mathbf{E}_0$ with two unary relation symbols $R[A]$ and $S[A]$ over the same attribute A. There are no constraints whatever. Let the view Π_R be that which retains the relation $R[A]$, but discards $S[A]$. The "obvious" complement to Π_R is Π_S, which preserves $S[A]$ and discards $R[A]$. Furthermore, update to Π_R with constant complement Π_S is exactly the "natural" update strategy – the state of $R[A]$ is changed at will, while the state of $S[A]$, which lies outside of the view, remains fixed.

[1] They will be re-expressed later in completely formal fashion as (upt:1) - (upt:5) of 3.1.

Unfortunately, Π_S is not the only complement of Π_R. Let Π_T be the view whose sole relation is $T[A]$. If M_R and M_S are the states of $R[A]$ and $S[A]$, respectively, then the state of $T[A]$ is the symmetric difference $(M_R \setminus M_S) \cup (M_S \setminus M_R)$. It is easy to see that Π_T is also a complement of Π_R, yet updates to Π_R with Π_T held constant are very different than those with Π_S held constant. Clearly, in a theory which looks for "natural" complements and their associated update strategies, Π_T should be excluded in favor of Π_S.

It might seem that such anomalous examples cannot arise in the context of projections on a single relation, such as in schema $\mathbf{E}_1$ above. However, although they are a bit peculiar, they certainly do exist. Returning to the schema $\mathbf{E}_1$ of 1.2, let $\mathsf{Dom}(C)$ denote the set of allowable domain values for attribute C, let $\alpha : \mathsf{Dom}(C) \to \mathsf{Dom}(C)$ be any permutation of $\mathsf{Dom}(C)$, and define the view $\Pi'_{BC} = (R'[BC], \pi'_{BC})$ as follows. States of Π'_{BC} are binary tuples on the domains of B and C, just as are states of the projection $R[BC]$. However, the tuples which are in the instance of $R'[BC]$ have their C-position values translated by α. Specifically, let M be a state of $\mathbf{E}_1$. For any $b \in \mathsf{Dom}(B)$, let $\#_A(b)$ denote the number of distinct values for attribute A associated with b in the relation $\pi_{AB}(M)$. Define $\pi'_{BC}(M) = \{(b,c) \mid (b,c) \in \pi_{BC}(M) \text{ and } \#_A(b) \text{ is odd}\} \cup \{(b,\alpha(c)) \mid (b,c) \in \pi_{BC}(M) \text{ and } \#_A(b) \text{ is even}\}$. It is easy to see that Π'_{BC} is a Π_B-complement of Π_{AB}, and the updates permitted to Π_{AB} under constant complement Π'_{BC} are exactly the same as those permitted under constant complement Π_{BC}. However, the translation is quite different. As a specific example, suppose that $\mathsf{Dom}(C) = \{c_0, c_1, c_2\}$, and let $\alpha(c_i) = c_{(i+1) \bmod 3}$. Let $M = \{(a_0, b_0, c_0), (a_1, b_1, c_1), (a_2, b_1, c_1)\}$. Then $\pi'_{BC}(M) = \{(b_0, c_0), (b_1, \alpha(c_1))\} = \{(b_0, c_0), (b_1, c_2)\}$. Now suppose that the tuple (a_1, b_0) is to be added to the state of Π_{AB}. Under constant complement Π_{BC}, the translation is simply to insert (a_1, b_0, c_0) to M. However, under constant complement Π'_{BC}, the translation is to insert the tuple $(a_1, b_0, \alpha^{-1}(c_0)) = (a_1, b_0, c_2)$, and to change the tuple (a_0, b_0, c_0) to $(a_0, b_0, \alpha^{-1}(c_0)) = (a_1, b_0, c_2)$, so that the state of E_1 after the update is $\{(a_0, b_0, c_2), (a_1, b_0, c_2), (a_1, b_1, c_1), (a_2, b_1, c_1)\}$. It is difficult to imagine a circumstance under which update to Π_{AB} with constant complement Π'_{BC} would be desirable. It is clear that Π_{BC} is *the* complement to be kept constant. Yet, the constant-complement approach itself gives no preference to Π_{BC} over Π'_{BC}.

The question might arise as to why a theory is needed to rule out complements such as Π_T and Π'_{BC}, which would never be chosen on their own merits over the desired complements Π_S and Π_{BC} in the first place. The answer is that the primary goal is not to rule out Π_T and Π'_{BC}, but rather to rule in Π_S and Π_{BC} as the only choices, on firm theoretical grounds. Although no "reasonable" alternatives come to mind, that is not a proof that none exist. Furthermore, in situations in which the base schema is much more complex, it may not be all that easy to evaluate, or even to find, suitable complements. Therefore, a firm theoretical foundation which selects complement views which induce the desired properties of closed update strategies would be very useful.

1.4 Addressing the uniqueness problem with order-based concepts The underlying theme in the results developed in this paper is that the addition of order-based concepts provides the key tool for establishing uniqueness of closed update strategies, as described in (cc:1)-(cc:4) of 1.2, thus excluding these anomalous cases. Specifically, database schemata are identified with the underlying sets of legal instances, and the set of legal instances of database schemata are taken to be ordered sets, with database mappings order-based morphisms. In the relational model, the ordering is defined by relation-by-relation inclusion; the basic operations of projection, selection, and join are monotonic with respect to this ordering. However, set difference is not. Under these conditions, the central result is for view updates which are order-based, in the sense that they are realizable as sequence of insertions and deletions, the way in which they are represented back in the base schema is independent of the choice of complement.

In the examples $\mathbf{E}_0$ and $\mathbf{E}_1$, it is easily seen that the database mappings associated with the "good" views Π_R, Π_S, Π_{AB}, and Π_{BC} are monotonic, while those associated with the "bad" views Π_T and Π'_{BC} are not. Thus, these undesirable views may be excluded if attention is restricted to views whose underlying mappings are monotonic.

The example schema $\mathbf{E}_0$ is taken from [15], in which a study of unique decompositions was made. In that paper, it was shown that Π_S is the only complement of Π_R which is monotonic. However, the results of [15] are restricted to *direct* complements, in which the views are independent, in the sense that there are no constraints which tie their states together. With reference to Fig. 1, the meet must be the empty view. Clearly, this is not a realistic assumption to make in the context of view update; indeed, the view Π_{AB} of schema $\mathbf{E}_1$ above has no direct complement. Thus, the results of [15] are not applicable even to so simple an example as schema $\mathbf{E}_1$ of 1.2.

It is not possible to extend the results of [15] to pairs of views which are not independent of one another. The tact taken in the present paper is rather to show that when the axioms (cc:1)-(cc:4) of 1.2 are extended with order-based properties, it is the order-based updates themselves, and not necessarily the complements, which are unique.

The results presented here are limited in that they apply only to updates which are insertions, deletions, or realizable as sequences of such. They do not apply, in general, to modifications. For example, let $\mathbf{E}_2$ be the relational schema which is identical to $\mathbf{E}_1$, save that the additional functional dependency $B \to A$ holds. With constant complement Π_{BC}, the only updates which are allowed to Π_{AB} are those which replace the A component of a tuple (a, b) with a new value. It is not possible to realize such an update as an insertion followed by a deletion, since the insertion would result in a violation of the functional dependency $B \to A$, which embeds in $R[AB]$. Thus, with the natural relation-by-relation inclusion, the theory developed here has nothing to say about updates to this view. It is an order-based theory, and updates which sidestep the order are not covered. Fortunately, there is still a way to establish uniqueness of the update strategy. In many cases, including this one, it is possible to find another order which will render these updates to be order-based, while retaining the monotonicity of the underlying view mappings. A solution for this example is presented in 4.6; however, a general theory of how to manage such updates must await a future paper.

1.5 An overview of this paper Sections 2 and 3 provide the fundamental background results necessary for the study of closed updates in an order-based context, while Sections 4, 5 and 6 provide the main results. Section 2 is a succinct presentation of the necessary database ideas within an order-based framework, while Section 3 extends the classical results of Bancilhon and Spyratos [3] to the order-based context. Section 4 contains the fundamental uniqueness result, showing that for order-based views, view updates defined by sequences of insertions and deletions are uniquely represented in the base schema. Section 5 looks in detail at the question of how an order-based update strategy may be extracted directly from a view, without the need to identify a complement explicitly. Section 6 applies the results of Section 5 to the classical setting of projective views on a single-relation schema constrained by equality-generating dependencies, and shows that under suitable circumstances, the classical complement defined by an extant join dependency defines the only possibility for an order-based update strategy. Finally, Section 7 provides some conclusions and further directions.

This is an extended version of the conference paper [16]. Some corrections and clarifications, as well as an example or two, have been added to Sections 2 and 3 of this paper, but otherwise they differ little from the corresponding sections of [16]. Section 4 of the current paper contains a newer and simpler proof of the main result 4.2, and adds some further examples (4.7 and 4.8) on non-combinability of update strategies. Sections 5 and 6 of the present paper are entirely new.

1.6 Applicability and relationship to other approaches to the view update problem Over the years, there has been a substantial amount of research on the topic of view updates, a majority of which has focused upon the relational model; some quite recent [5]. The problem has also been studied, to a limited extent, in the context of the ER model [20, Sec. 8.3] and the object-oriented model [4]. However, except for the seminal work of Bancilhon and Spyratos [3] and that of the author [12], most of this research has the flavor of open strategies.

The work reported here depends only upon the data model possessing a natural order structure, and so is potentially applicable to any of the contexts identified above. Although the examples used in this paper are all relational, the theory is not tied to that model in any way. Nonetheless, it does presume a situation in which the state of the database is represented by a single structure, rather than as a set of constraints, so applicability to deductive models is not clear. In any case, research on updates to deductive databases has taken a direction quite its own; see [18] for a recent survey.

As mentioned in 1.4, the choice of order as a tool to establish uniqueness results is motivated by earlier work [15], in which such tools were used to show that decompositions into independent components are unique when databases are suitably ordered. However, the techniques employed in this paper are vastly different than those of [15], since independence is not a property which most user views have. This paper is in some way part of a long overdue sequel to [12], which laid out principles for the support of closed views, but never unified the results to fruition.

1.7 The results and the relational model Although the results are not specific to the relational model, it remains not only the most widely used data model, but the one with, by far, the most extensive theoretical foundation. Therefore, it is important to show applicability to that model. To this end, several examples set within the classical relational context which illustrate the use of the results developed here are presented. On the other hand, because this paper is not about the relational model, it would be inappropriate to provide a detailed summary of its features, particularly because they are so widely known. Thus, it must be assumed that the reader is familiar with the standard terminology, notation, and results within that context. The monograph [1] provides a relatively recent survey and the necessary background.

The sole exception is Section 6, which is devoted exclusively to a focused case of the relational model. There, because some of the detailed results which are used are not widely known, they are summarized in enough detail to provide continuity of the presentation.

1.8 Basic notation and background In addition to concepts from database theory, this paper makes extensive use of terminology and results from order theory. Although some essential results are summarized in 2.1, a more general conceptual knowledge is expected. A good reference in this regard is [6].

At a more fundamental level, sequences of items will often be surrounded by angle brackets; *e.g.*, $\langle a, b, c, d, e\rangle$. In that case, for a sequence σ, $\mathsf{First}(\sigma)$ is the first element of σ, and $\mathsf{Last}(\sigma)$ the last. Thus, if σ is the example sequence given above, $\mathsf{First}(\sigma) = a$ and $\mathsf{Last}(\sigma) = e$. The power set of a set S; *i.e.*, the set of all subsets of S, is denoted $\mathbf{2}^S$. Finally, the symbols $\mathbb{N}$ and $\mathbb{Z}$ have their usual meanings as the sets of natural numbers and integers, respectively.

2. Database Concepts in the Order-Based Context

In this section, the fundamental ideas of database schemata, morphisms, views, and complements are formulated within the context of order. That is, database schemata are presumed to have an underlying order to their states, and database mappings are assumed to preserve this order. A few of the ideas are related to those presented in [15]; however, the details are quite different, since that work dealt with a much more specialized context.

2.1 Posets Familiarity with the fundamental ideas of posets, such as presented in [6], is presumed; only a few notational and terminological points are reviewed here. A *partially ordered set* (*poset*) is a pair $\mathbf{P} = (P, \leq)$ in which P is a set and $\leq$ is a reflexive, transitive, and antisymmetric relation on P. Given posets $\mathbf{P} = (P, \leq)$ and $\mathbf{Q} = (Q, \leq)$, 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 $p_1 \leq p_2$ implies that $f(p_1) \leq f(p_2)$. The morphism f is *open* if, for any $q_1, q_2 \in Q$ with $q_1 \leq q_2$, there are $p_1 \in f^{-1}(q_1)$, $p_2 \in f^{-1}(q_2)$ with $p_1 \leq p_2$. In other words, f is open if Q carries the least order which renders f a morphism. Following the standard terminology of category theory [2, 3.8], 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.

2.2 Schemata and morphisms Mathematically, a database schema with order is just a partially ordered set, and a morphism of such schemata is a poset morphism. However, to emphasize the database aspects, a special notation is employed in this work. Specifically, 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 $\mathbf{D}$.

In the case that the order $\leq_{\mathbf{D}}$ is the identity order in which $M \leq_{\mathbf{D}} N$ iff $M = N$, $\mathbf{D}$ is called a *flat* or *unordered* schema. In this case, the order relation $\leq_{\mathbf{D}}$ plays no rôle whatever, and the schemata and morphisms are just sets and functions, respectively.

To simplify terminology, throughout the rest of this paper, the term *schema* shall mean *schema with order*, unless specifically stated to the contrary.

Given database schemata $\mathbf{D}_1 = (\mathsf{LDB}(\mathbf{D}_1), \leq_{\mathbf{D}_1})$ and $\mathbf{D}_2 = (\mathsf{LDB}(\mathbf{D}_2), \leq_{\mathbf{D}_2})$, a *morphism* $h : \mathbf{D}_1 \to \mathbf{D}_2$ is a function $h : \mathsf{LDB}(\mathbf{D}_1) \to \mathsf{LDB}(\mathbf{D}_2)$ which is a poset morphism with respect to the orders $\leq_{\mathbf{D}_1}$ and $\leq_{\mathbf{D}_2}$. It is called an *open surjection* precisely in the case that it has that property as a poset morphism.

2.3 Views and Congruences An *order view* of the schema $\mathbf{D}$ is a pair $\Gamma = (\mathbf{V}, \gamma)$ in which $\mathbf{V}$ is a schema and $\gamma : \mathbf{D} \to \mathbf{V}$ is an open surjection. Given order views $\Gamma_1 = (\mathbf{V}_1, \gamma_1)$ and $\Gamma_2 = (\mathbf{V}_2, \gamma_2)$ of $\mathbf{D}$, a *morphism* $h : \Gamma_1 \to \Gamma_2$ is a schema morphism $h : \mathbf{V}_1 \to \mathbf{V}_2$ with the property that $h \circ \gamma_1 = \gamma_2$, that is, such that the diagram to the right commutes. Since the morphisms γ_1 and γ_2 are open surjections, so too is h. The order views Γ_1 and Γ_2 are *isomorphic* just in case they satisfy the standard categorical notion of isomorphism [2, 3.8]; that is, there are morphisms $h_1 : \Gamma_1 \to \Gamma_2$ and $h_2 : \Gamma_2 \to \Gamma_1$ such that $h_1 \circ h_2$ and $h_2 \circ h_1$ are both identities. The *congruence* of order view $\Gamma = (\mathbf{V}, \gamma)$ is the equivalence relation $\mathsf{Congr}(\Gamma)$ on $\mathsf{LDB}(\mathbf{D})$ defined by $(M, N) \in \mathsf{Congr}(\Gamma)$ iff $\gamma(M) = \gamma(N)$. Given $M \in \mathsf{LDB}(\mathbf{D})$, the notation $[M]_\Gamma$ is shorthand for the more cumbersome $[M]_{\mathsf{Congr}(\Gamma)}$. Both denote the equivalence class of M under the equivalence relation $\mathsf{Congr}(\Gamma)$.

$$\begin{tikzcd} & \mathbf{D} \arrow[dl, "\gamma_1"'] \arrow[dr, "\gamma_2"] & \\ \mathbf{V}_1 \arrow[rr, "h"] & & \mathbf{V}_2 \end{tikzcd}$$

Throughout the remainder of this work, unless specifically stated to the contrary, the term *view* will mean *order view*, as defined above.

Before proceeding further, it is important to establish that the general notions introduced here are applicable to the classical relational setting. The result 2.5 below shows that the common *Select-Project-Join* mappings of the relational theory define order based views in the sense of 2.3 above. First, some clarification of terminology and notation is in order.

2.4 Order-based schemata and morphisms in the classical relational context The *named perspective* [1, Sec. 3.2] is used; this means that the columns of relations are identified by attribute names from a finite, nonempty universe $\mathbf{U}$. For a given attribute A, $\mathsf{Dom}(A)$ denotes the set of all allowable values for that attribute. A *relational schema* $\mathbf{D}$ consists of a finite set of relational symbols $\mathsf{Rel}(\mathbf{D})$, each with a finite arity $\mathsf{Arity}(R) \subseteq \mathbf{U}$. An *unconstrained database* M over a relational schema $\mathbf{D}$ consists of a set of relations $\{M^R \mid R \in \mathsf{Rel}(\mathbf{D})\}$ of the appropriate arities. The set of all unconstrained databases on $\mathbf{D}$ is denoted $\mathsf{DB}(\mathbf{D})$. The *natural ordering* $\subseteq_{\mathbf{D}}$ on the databases of $\mathbf{D}$ is defined via relation-by-relation inclusion; *i.e.*, $M_1 \subseteq_{\mathbf{D}} M_2$ iff ${M_1}^R \subseteq {M_2}^R$ for all relation symbols R of $\mathbf{D}$.

Relational schemata are commonly constrained by Horn sentences. If only universal quantifiers are allowed, such constraints are termed *full dependencies*; if existential quantifiers are allowed on positive atoms, they are termed *embedded dependencies* [1, Chap. 10]. Functional dependencies (*FD*'s) and join dependencies (*jd*'s) are full dependencies. If there is a set Φ of full (resp. embedded) dependencies such that $\mathsf{LDB}(\mathbf{D}) = \{M \in \mathsf{DB}(\mathbf{D}) \mid M \models \mathsf{Dep}(\mathbf{D})\}$, then $\mathbf{D}$ is said to be *constrained by full* (resp. *embedded*) *depen-*

dencies; in both cases the notation $\mathsf{Dep}(\mathbf{D})$ is used to denote Φ.

It is useful to know that the most common types of morphisms within the relational framework define views in the order-based context. The database mappings which will be considered in this work are those which are built up from compositions of the primitives *selection*, *projection*, *join*, and *renaming*. Such compositions are termed *SPJR-morphisms*. [1, Sec. 4.4]. It is clear that *SPJR*-mappings are monotonic with respect to the natural orderings, and so define morphisms in the order-based sense. It is equally important to note that database mappings which involve negation, such as difference and division, are not monotonic.

2.5 Proposition — *SPJR*-morphisms define views *Let* $\mathbf{D}$ *and* $\mathbf{V}$ *be relational schemata, and suppose furthermore that* $\mathbf{D}$ *is constrained by full dependencies. Let* $\gamma : \mathbf{D} \to \mathbf{V}$ *be a surjective SPJR-morphism. Then* $\mathbf{V}$ *is also constrained by full dependencies, and* γ *is an open poset morphism with respect to the natural orderings* $\subseteq_{\mathbf{D}}$ *and* $\subseteq_{\mathbf{V}}$*, so that* $\Gamma = (\mathbf{V}, \gamma)$ *is a view of* $\mathbf{D}$ *in the order-based sense.*

PROOF OUTLINE: First of all, that the class of full dependencies is closed under projection and join is established in [17, Thm. 5.4]. Selection may be added easily using the submodel characterization of universal Horn sentences [19, Thm. 25.13]. Thus, $\mathbf{V}$ is also constrained by full dependencies.

To establish that γ is open, break the problem into three cases, one each for projection, selection, and join, each with possible renaming. Let $N_1, N_2 \in \mathsf{LDB}(\mathbf{V})$ with $N_1 \subseteq_{\mathbf{V}} N_2$. Now, reflect N_1 and N_2 back into $\mathbf{D}$ as minimal structures P_1 and P_2, respectively, possibly with variables, and apply the chase inference procedure [1, Sec. 10.2] to generate legal databases of $\mathbf{D}$. When variables occur (this will happen with projection), they must be the same in tuples which are in both N_1 and N_2, and their eventual bindings to domain values must be the same in each case. Let $\hat{P}_1, \hat{P}_2 \in \mathsf{LDB}(\mathbf{D})$ be the models generated from the chase on P_1 and P_2, respectively. Then $\hat{P}_1 \subseteq_{\mathbf{D}} \hat{P}_2$, with $\gamma(\hat{P}_1) = N_1$ and $\gamma(\hat{P}_2) = N_2$. □

2.6 Relational projections The key relational examples used in this work are projections. Although this framework has already been used in the introduction, it is important to crystallize the notation for the formal part of the paper. Let $\mathbf{D}$ be a relational schema which is constrained by full dependencies, let $R \in \mathsf{Rel}(\mathbf{D})$, let $\mathbf{A} = \mathsf{Arity}(R)$, and let $\mathbf{B} \subseteq \mathbf{A}$. The $\mathbf{B}$-*projection* of R is the view $\Pi_{\mathbf{B}} = (R[\mathbf{B}], \pi_{\mathbf{B}})$ with $R[\mathbf{B}]$ the schema with a single relational symbol R on attribute set $\mathbf{B}$, and $\pi_{\mathbf{B}} : \mathbf{D} \to R[\mathbf{B}]$ the morphism which sends $M \in \mathsf{LDB}(\mathbf{D})$ to the projection onto attributes in $\mathbf{B}$ of the relation R in $\mathbf{D}$. $\mathsf{Dep}(R[\mathbf{B}])$ is taken to be the set of full dependencies which render $\pi_{\mathbf{A}}$ an open surjection. In view of 2.5 above, $\mathsf{Dep}(R[\mathbf{B}])$ is guaranteed to exist, and $\Pi_{\mathbf{B}}$ is guaranteed to be a view in the order-based sense.

The focus now returns to the more general context. The following proposition characterizes views and their isomorphism classes in terms of their congruences. The proof is completely straightforward, and so omitted.

2.7 Proposition *Let $\mathbf{D}$ be a schema, and let $\Gamma_1 = (\mathbf{V}_1, \gamma_1)$ and $\Gamma_2 = (\mathbf{V}_2, \gamma_2)$ be views of $\mathbf{D}$.*

(a) *Every view morphism $\Gamma_1 \to \Gamma_2$ is an open surjection of posets.*

(b) *There is at most one morphism $\Gamma_1 \to \Gamma_2$.*

(c) *There is a morphism $\Gamma_1 \to \Gamma_2$ iff $\mathsf{Congr}(\Gamma_1) \subseteq \mathsf{Congr}(\Gamma_2)$.*

(d) *The views Γ_1 and Γ_2 are isomorphic iff $\mathsf{Congr}(\Gamma_1) = \mathsf{Congr}(\Gamma_2)$.* $\square$

2.8 View equivalence and order The (equivalence) class of all views which are isomorphic to Γ is denoted $[\Gamma]$. In view of 2.7(d) above, $[\Gamma_1] = [\Gamma_2]$ iff $\mathsf{Congr}(\Gamma_1) = \mathsf{Congr}(\Gamma_2)$. The notation $[\Gamma_2] \leq [\Gamma_1]$ denotes that there is a morphism $\Gamma_1 \to \Gamma_2$, or, equivalently, that $\mathsf{Congr}(\Gamma_1) \subseteq \mathsf{Congr}(\Gamma_2)$. As a shorthand, $\Gamma_2 \leq \Gamma_1$ shall also be written to denote this fact, with the understanding that $\leq$ is not a partial order on the views themselves, but only on the underlying equivalence classes.

If $\Gamma_1 = (\mathbf{V}_1, \gamma_1)$ and $\Gamma_2 = (\mathbf{V}_2, \gamma_2)$ are views of the schema $\mathbf{D}$ with $\Gamma_2 \leq \Gamma_1$, then the unique morphism $\Gamma_1 \to \Gamma_2$ is denoted $\lambda\langle\Gamma_1, \Gamma_2\rangle$. In this case, $\mathbf{V}_2$ may be regarded as a view of $\mathbf{V}_1$ under the view mapping $\lambda\langle\Gamma_1, \Gamma_2\rangle$. Specifically, the *relativization* of Γ_2 to Γ_1 is the view $\Lambda(\Gamma_1, \Gamma_2) = (\mathbf{V}_2, \lambda\langle\Gamma_1, \Gamma_2\rangle)$ of $\mathbf{V}_1$.

2.9 Order-compatible congruences In the context of a flat schema $\mathbf{D}$ with no order relation, every equivalence relation R on $\mathsf{LDB}(\mathbf{D})$ gives rise to a view whose states are the equivalence classes of R, with the view mapping the natural projection of an element to its equivalence class. In the context of order-based schemata, an additional constraint mandating that the equivalence respect the order on the database states must be imposed.

Specifically, let $\mathbf{D}$ be an order-based schema, and let R be an equivalence relation on $\mathsf{LDB}(\mathbf{D})$. Call R *order compatible* for $\mathbf{D}$ if for every pair $(M_1, M_2) \in R$ with $M_1 \leq_{\mathbf{D}} M_2$ and every $M_3 \in \mathsf{LDB}(\mathbf{D})$ with $M_1 \leq_{\mathbf{D}} M_3 \leq_{\mathbf{D}} M_2$, $(M_1, M_3) \in R$ as well. Now, define $\Theta_R = (\mathsf{LDB}(\mathbf{D})/R, \theta_R)$ to be the view of $\mathbf{D}$ with $\mathsf{LDB}(\mathbf{D})/R$ the schema whose underlying set is $\mathsf{LDB}(\mathbf{D})/R$, the set of equivalence classes of R, with $\theta_R : \mathbf{D} \to \mathsf{LDB}(\mathbf{D})/R$ the morphism which sends each $M \in \mathsf{LDB}(\mathbf{D})$ to its equivalence class $[M]_R$ under R. The order $\leq_{\mathbf{\Theta_R}}$ is given by $[M]_R \leq_{\mathbf{\Theta_R}} [N]_R$ iff $(\exists M_1 \in [M]_R)(\exists N_1 \in [N]_R)(M_1 \leq_{\mathbf{D}} N_1)$. The order compatibility of R ensures that $\leq_{\mathbf{\Theta_R}}$ is a partial order, and the construction itself ensures that θ_R is an open surjection. Thus, Θ_R is indeed an order-based view of $\mathbf{D}$.

Note, conversely, that for any view $\Gamma = (\mathbf{V}, \gamma)$, $\mathsf{Congr}(\Gamma)$ is an order-based congruence, since otherwise the order $\leq_{\mathbf{V}}$ would not be well defined.

2.10 Products and complements of views Let $\Gamma_1 = (\mathbf{V}_1, \gamma_1)$ and $\Gamma_2 = (\mathbf{V}_2, \gamma_2)$ be views of the schema $\mathbf{D}$. The product $\Gamma_1 \times \Gamma_2 = (\mathbf{V}_1 \,{}_{\gamma_1}\!\!\otimes_{\gamma_2} \mathbf{V}_2, \gamma_1 \otimes \gamma_2)$ has $\mathsf{LDB}(\mathbf{V}_1 \,{}_{\gamma_1}\!\!\otimes_{\gamma_2} \mathbf{V}_2) = \{(\gamma_1(M), \gamma_2(M)) \mid M \in \mathsf{LDB}(\mathbf{D})\}$, with the induced product ordering; *i.e.*, $(N_1, N_2) \leq_{\mathbf{V}_1 \,{}_{\gamma_1}\!\otimes_{\gamma_2} \mathbf{V}_2} (N_1', N_2')$ iff $N_1 \leq_{\mathbf{V}_1} N_1'$ and $N_2 \leq_{\mathbf{V}_2} N_2'$. The morphism $\gamma_1 \otimes \gamma_2 : \mathbf{D} \to \mathbf{V}_1 \,{}_{\gamma_1}\!\!\otimes_{\gamma_2} \mathbf{V}_2$ is given on elements by $M \mapsto (\gamma_1(M), \gamma_2(M))$. Example 2.11 below shows that $\gamma_1 \otimes \gamma_2$ need not be open (although it is always a surjective poset morphism), so $\Gamma_1 \times \Gamma_2$ need not be a view.

The pair $\{\Gamma_1, \Gamma_2\}$ of views is said to form a *subdirect complementary pair*, just in case $\gamma_1 \otimes \gamma_2 : \mathbf{D} \to \mathbf{V}_1 \,{}_{\gamma_1}\!\!\otimes_{\gamma_2} \mathbf{V}_2$ is a poset isomorphism. Note that in this case $\Gamma_1 \times \Gamma_2$ is always a view, which is in fact isomorphic to the identity view. In this case, it is also said that Γ_1 and Γ_2 are *subdirect complements* of one another.

As mentioned above, it is entirely possible for $\gamma_1 \otimes \gamma_2$ to be a bijection and a poset morphism without being a poset isomorphism, since the order on $\mathbf{V}_1 \,{}_{\gamma_1}\!\!\otimes_{\gamma_2} \mathbf{V}_2$ induced by the product order on $\mathbf{V}_1 \times \mathbf{V}_2$ is, in general, strictly stronger than that induced by the morphism $\gamma_1 \otimes \gamma_2$. Thus, two order-based views may be complementary in a "flat" environment in which order is ignored, without being complements in the order-based sense. The following example illustrates this possibility.

2.11 Example — views not closed under product Let $\mathbf{E}$ be the schema with $\mathsf{LDB}(\mathbf{E}) = \{a, b, c, d\}$, $b \leq_{\mathbf{E}} a$, and no other order on the elements. Define the views $\Omega_1 = (\mathbf{W}_1, \omega_1)$ with $\mathsf{LDB}(\mathbf{W}_1) = \{ac, bd\}$ and $bd \leq ac$, and $\Omega_2 = (\mathbf{W}_2, \omega_2)$ with $\mathsf{LDB}(\mathbf{W}_2) = \{ad, bc\}$ and $bc \leq ad$. The view morphisms ω_1 and ω_2 are defined by the table to the right, and are easily seen to be open surjections. On $\mathbf{W}_1 \,{}_{\omega_1}\!\!\otimes_{\omega_2} \mathbf{W}_2$, $(bd, bc) \leq (ac, bc) \leq (ac, ad)$ and $(bd, bc) \leq (bd, ad) \leq (ac, ad)$, which is strictly stronger than the order induced from $\mathbf{E}$. Thus, while $\omega_1 \otimes \omega_2$ is a bijection and a poset morphism, it is not an isomorphism. Hence $\{\Omega_1, \Omega_2\}$ is not a complementary pair in the order-based sense, even though it is in the set-based sense.

x	$\omega_1(x)$	$\omega_2(x)$
a	ac	ad
b	bd	bc
c	ac	bc
d	bd	ad

In the examples of 1.2 and 1.3, the common view Π_B was referred to as the *meet* of Π_{AB} and Π_{BC}. This notion of the common component of two views is extremely important in a number of contexts involving views and database decomposition, including the generalization of acyclic decompositions [14] and decomposition into independent components [15]. Not surprisingly, it is also central to the constant complement strategy. For it to be well defined, the congruences of the two views must commute; the definition follows.

2.12 Fully commuting views and meet complements The pair $\{\Gamma_1, \Gamma_2\}$ of views of $\mathbf{D}$ is called a *fully commuting pair* if $\mathsf{Congr}(\Gamma_1) \circ \mathsf{Congr}(\Gamma_2) = \mathsf{Congr}(\Gamma_2) \circ \mathsf{Congr}(\Gamma_1)$, with "$\circ$" denoting ordinary relational composition. A subdirect complementary pair $\{\Gamma_1, \Gamma_2\}$ which is fully commuting is called a *meet-complementary pair*, and Γ_1 and Γ_2 are called *meet complements* of one another.

2.13 Generalized dependencies Let $\{\Gamma_1, \Gamma_2\}$ be a subdirect complementary pair.

(a) The $\{\Gamma_1, \Gamma_2\}$-*reconstruction dependency* on $\mathbf{V}_1 \,{}_{\gamma_1}\!\!\otimes_{\gamma_2} \mathbf{V}_2$, denoted $\otimes[\Gamma_1, \Gamma_2]$, is satisfied iff for any $M_1, N_1 \in \mathsf{LDB}(\mathbf{V}_1)$ and $M_2, N_2 \in \mathsf{LDB}(\mathbf{V}_2)$, if any three of the elements of the set $\{(M_1, M_2), (M_1, N_2), (N_1, M_2), (N_1, N_2)\}$ is in $\mathsf{LDB}(\mathbf{V}_1 \,{}_{\gamma_1}\!\!\otimes_{\gamma_2} \mathbf{V}_2)$, then so too is the fourth.

(b) Let $\Gamma_3 = (\mathbf{V}_3, \gamma_3)$ be a view of $\mathbf{D}$, with $[\Gamma_3] \leq [\Gamma_1]$ and $[\Gamma_3] \leq [\Gamma_2]$. The Γ_3-*independence dependency* on $\mathbf{V}_1 \,{}_{\gamma_1}\!\!\otimes_{\gamma_2} \mathbf{V}_2$, denoted $\otimes_{\Gamma_3}$, is satisfied iff for any $M_1 \in \mathsf{LDB}(\mathbf{V}_1)$ and $M_2 \in \mathsf{LDB}(\mathbf{V}_2)$, $((M_1, M_2) \in \mathsf{LDB}(\mathbf{V}_1 \,{}_{\gamma_1}\!\!\otimes_{\gamma_2} \mathbf{V}_2)) \Leftrightarrow (\lambda\langle\Gamma_1, \Gamma_3\rangle(M_1) = \lambda\langle\Gamma_2, \Gamma_3\rangle(M_2))$.

The $\{\Gamma_1, \Gamma_2\}$-reconstruction dependency above is a generalization of the classical join dependency from relational database theory, based upon the idea that items in the matching columns "pair off." It is a special case of a representation of subdirect products in universal algebra [11, §20, Thm. 2]. The Γ_3 independence dependency is an alternate generalization of a join dependency, embodying the idea that those tuples which are joined are precisely the ones which match on the common columns.

The following characterization first appeared in [12, 1.13]. Unfortunately, due to space constraints, no proof was presented. Because of its importance, the proof is sketched here. Note that this result is essentially independent of additional constraints imposed by the order-based context.

2.14 Theorem — characterization of meet-complementary pairs *Let $\{\Gamma_1, \Gamma_2\}$ be a subdirect complementary pair. Then the following conditions are equivalent.*

(a) $\{\Gamma_1, \Gamma_2\}$ *is a meet-complementary pair.*

(b) $\mathsf{Congr}(\Gamma_1) \circ \mathsf{Congr}(\Gamma_2)$ *is an equivalence relation.*

(c) $\mathbf{V}_1 {}_{\gamma_1}\!\!\otimes_{\gamma_2} \mathbf{V}_2$ *satisfies* $\otimes[\Gamma_1, \Gamma_2]$.

(d) $\mathbf{V}_1 {}_{\gamma_1}\!\!\otimes_{\gamma_2} \mathbf{V}_2$ *satisfies* $\otimes_{\Gamma_3}$, *with* Γ_3 *the view (unique up to equivalence) whose congruence is the smallest equivalence relation on* $\mathsf{LDB}(\mathbf{D})$ *containing both* $\mathsf{Congr}(\Gamma_1)$ *and* $\mathsf{Congr}(\Gamma_2)$.

PROOF: The implications (a) $\Leftrightarrow$ (b) and (a) $\Rightarrow$ (c) are straightforward and left to the reader, while (a) $\Leftrightarrow$ (d) is a special case of the Chinese-remainder characterization for schema decomposability [14, 2.1.5]. To show the implication, (c) $\Rightarrow$ (a) let $(M, N) \in \mathsf{Congr}(\Gamma_1) \circ \mathsf{Congr}(\Gamma_2)$. Then, there is a $P \in \mathsf{LDB}(\mathbf{D})$ with $(M, P) \in \mathsf{Congr}(\Gamma_1)$ and $(P, N) \in \mathsf{Congr}(\Gamma_2)$. Let (M_1, M_2), (N_1, N_2), and (P_1, P_2) denote the images, under $\gamma_1 \otimes \gamma_2$, of M, N, and P, respectively. Since $(M, P) \in \mathsf{Congr}(\Gamma_1)$, $M_1 = P_1$. Similarly, since $(P, N) \in \mathsf{Congr}(\Gamma_2)$, $N_2 = P_2$. Thus, $(P_1, P_2) = (M_1, N_2)$. In particular, $(M_1, N_2) \in \mathsf{LDB}(\mathbf{V}_1 {}_{\gamma_1}\!\!\otimes_{\gamma_2} \mathbf{V}_2)$. So, applying $\otimes[\Gamma_1, \Gamma_2]$ to the set $\{(M_1, M_2), (M_1, N_2), (N_1, N_2)\}$, it follows that $(N_1, M_2) \in \mathsf{LDB}(\mathbf{V}_1 {}_{\gamma_1}\!\!\otimes_{\gamma_2} \mathbf{V}_2)$. Thus, $((\gamma_1 \otimes \gamma_2)^{-1}(M_1, M_2),\ (\gamma_1 \otimes \gamma_2)^{-1}(N_1, M_2)) \in \mathsf{Congr}(\Gamma_2)$ and $((\gamma_1 \otimes \gamma_2)^{-1}(N_1, M_2), (\gamma_1 \otimes \gamma_2)^{-1}(N_1, N_2)) \in \mathsf{Congr}(\Gamma_1)$, so that $(M, N) \in \mathsf{Congr}(\Gamma_2) \circ \mathsf{Congr}(\Gamma_1)$, whence $\mathsf{Congr}(\Gamma_1) \circ \mathsf{Congr}(\Gamma_2) \subseteq \mathsf{Congr}(\Gamma_2) \circ \mathsf{Congr}(\Gamma_1)$. The opposite inclusion is proved in an analogous fashion, so that $\mathsf{Congr}(\Gamma_1) \circ \mathsf{Congr}(\Gamma_2) = \mathsf{Congr}(\Gamma_2) \circ \mathsf{Congr}(\Gamma_1)$. □

2.15 Meets and Γ-complements Let $\Gamma_1 = (\mathbf{V}_1, \gamma_1)$ and $\Gamma_2 = (\mathbf{V}_2, \gamma_2)$ be views of $\mathbf{D}$ which form a subdirect complementary pair. In the case that the equivalent conditions of 2.14 above are satisfied, the view Γ_3 guaranteed by (d) is called the *meet* of $\{\Gamma_1, \Gamma_2\}$, and is denoted $\Gamma_1 \wedge \Gamma_2 = (\mathbf{V}_1 {}_{\gamma_1}\!\!\wedge_{\gamma_2} \mathbf{V}_2, \gamma_1 \wedge \gamma_2)$. The set $\{\Gamma_1, \Gamma_2\}$ is then said to form a *meet-complementary pair*, or Γ_3-*complementary pair*, and Γ_1 and Γ_2 are called Γ_3-*complements* of one another.

The meet view Γ_3 is defined only up to isomorphism of views, but this is no limitation for the work presented here. Also, since $\mathbf{V}_1 {}_{\gamma_1}\!\!\wedge_{\gamma_2} \mathbf{V}_2$ factors through both Γ_1 and Γ_2, it is immediate that the order on $\mathsf{LDB}(\mathbf{V}_1 {}_{\gamma_1}\!\!\wedge_{\gamma_2} \mathbf{V}_2)$ induced by either $\leq_{\mathbf{V}_1}$ or $\leq_{\mathbf{V}_2}$ is the same as that induced by $\leq_{\mathbf{D}}$, so there is no question that this order is well defined.

2.16 Products and meets in the relational context Let $\mathbf{D}$ be a relational schema, and let $\Gamma_1 = (\mathbf{V}_1, \gamma_1)$ and $\Gamma_2 = (\mathbf{V}_2, \gamma_2)$ be views of $\mathbf{D}$ defined by SPJR-morphisms. The relation names of $\mathbf{V}_1 {}_{\gamma_1}\!\otimes_{\gamma_2} \mathbf{V}_2$ will be the disjoint union of those in $\mathbf{V}_1$ and $\mathbf{V}_2$. Furthermore, if the pair $\{\Gamma_1, \Gamma_2\}$ forms a subdirect decomposition, then, in general, there will be interrelational constraints. For example, let $\mathbf{E}_1$ be the schema introduced in 1.2, with the single relation symbol $R[ABC]$, governed by the FD $B \to C$. Let $\Gamma_1 = \Pi_{AB}$ and $\Gamma_2 = \Pi_{BC}$. On $\mathbf{V}_1 {}_{\gamma_1}\!\otimes_{\gamma_2} \mathbf{V}_2$, which contains relation symbols $R_1[AB]$ and $R_2[BC]$ (the subscripts on the R's are used here to avoid name collision), the FD $B \to C$ holds on $R_2[BC]$, and the jd $R_1[AB] \bowtie R_2[BC]$ binds the two relations together. In this case, it is easy to see that $\{\Gamma_1, \Gamma_2\}$ are Π_B-complements, as condition (d) of 2.14 applies. On the other hand, if the FD $A \to C \in \mathsf{Dep}(\mathbf{D})$ as well, as in the schema $\mathbf{E}_2$ of 1.3, then while $\{\Gamma_1, \Gamma_2\}$ remains a subdirect complementary pair, it is no longer a meet-complementary pair, as condition (d) of 2.14 fails [12, 1.11].

The generalization of this idea makes use of the idea of dependency preservation. A decomposition $\{\Gamma_1 = (\mathbf{V}_1, \gamma_1), \Gamma_2 = (\mathbf{V}_2, \gamma_2)\}$ of a relational schema $\mathbf{D}$ is *dependency preserving* if $\mathsf{Dep}(\mathbf{D})$ is recoverable from the reconstruction dependency (usually a join dependency), together with $\mathsf{Dep}(\mathbf{V}_1) \cup \mathsf{Dep}(\mathbf{V}_2)$. If $\mathsf{Dep}(\mathbf{D})$ consists of FD's, this means that a cover of $\mathsf{Dep}(\mathbf{D})$ embeds in the relations of the view schemata. For details, see [1, Sec. 11.2]. The formalization is then the following.

2.17 Proposition — meets of projections in the relational context *Let $\mathbf{D}$ be a relational schema consisting of a single relation $R[\mathbf{U}]$, and let $\mathbf{A}, \mathbf{B} \subseteq \mathbf{U}$. Assume further that $\mathsf{Dep}(\mathbf{D})$ is generated by full dependencies, and that $\bowtie [\mathbf{A}, \mathbf{B}] \in \mathsf{Dep}(\mathbf{D})$. Then the pair $\{\Pi_{\mathbf{A}}, \Pi_{\mathbf{B}}\}$ forms a meet-complementary pair iff the decomposition is dependency preserving. In this case, $\Pi_{\mathbf{A}} \wedge \Pi_{\mathbf{B}} = \Pi_{\mathbf{A} \cap \mathbf{B}}$.*

PROOF OUTLINE: It is straightforward to verify that condition (d) of 2.14 is satisfied for these projections. □

3. Updates in the Order-Based Context

Using the context developed in the previous section, the ideas of constant-complement update within an order-based framework are now established. These results extend those of Bancilhon and Spyratos [3] to the order-based case. Because this extension is far from trivial, and because the formalism employed here is different than that of [3], all results are proven directly. The results presented here strengthen those of [3], not only in extension to the order-based case, but also in that meet complementation is used to provide a complete bijective correspondence between update strategies and complements.

3.1 Update strategies Let $\mathbf{D}$ be a database schema. A *closed update family* for $\mathbf{D}$ is an order-compatible equivalence relation U on $\mathsf{LDB}(\mathbf{D})$. Think of a pair $(M_1, M_2) \in U$ as describing an update of the database from state M_1 to state M_2. The equivalence-relation requirement implies that the identity update is always allowable, that all updates are reversible, and that updates may be composed.

Now let $\Gamma = (\mathbf{V}, \gamma)$ be a view of $\mathbf{D}$, and let T be a closed update family for $\mathbf{V}$. Formally, an *update strategy* for T with respect to U is a partial function $\rho : \mathsf{LDB}(\mathbf{D}) \times \mathsf{LDB}(\mathbf{V}) \to \mathsf{LDB}(\mathbf{D})$ which has the eight properties (upt:1)-(upt:8) listed below. The idea of an update strategy is illustrated in the diagram to the right. Put another way, $\rho : (\mathsf{current\ state\ of\ }\mathbf{D}, \mathsf{new\ state\ of\ }\mathbf{V}) \mapsto \mathsf{new\ state\ of\ }\mathbf{D}$ in such a way that the new state of $\mathbf{D}$ gives rise to the desired new state of $\mathbf{V}$. The update to $\mathbf{V}$ must lie in T, and the reflected update to $\mathbf{D}$ must lie in U. In practice, U is often taken to be all possible updates on $\mathbf{D}$, *i.e.*, $\mathsf{LDB}(\mathbf{D}) \times \mathsf{LDB}(\mathbf{D})$, but this is in no way essential to the theory which follows.

$$\begin{tikzcd}
M_1 \arrow[r, mapsto, dashed, "\text{reflected update}"] \arrow[d, mapsto, "\gamma"'] & \rho(M_1, N_2) \arrow[d, mapsto, "\gamma"] \\
\gamma(M_1) = N_1 \arrow[r, mapsto, "\text{view update}"] & N_2 = \gamma(\rho(M_1, N_2))
\end{tikzcd}$$

Of the eight conditions listed below, the first five, (upt:1)-(upt:5), restate in more formal terms (cc:1)-(cc:4) of 1.2, and so constitute an alternative formulation of those in the original work [3]. Specifically, (upt:1) and (upt:2) formalize (cc:1), while (upt:3), (upt:4), and (upt:5) formalize (cc:2), (cc:4), and (cc:3), respectively. The last three, (upt:6)-(upt:8), are specific to the order-based context. The notation $\rho(M,N)\!\downarrow$ means that ρ is defined on the argument (M,N). If a formula involving ρ appears in a formula, then it is implicitly assumed that it is defined.

(upt:1) $\rho(M,N)\!\downarrow$ iff $(\gamma(M), N) \in T$.

(upt:2) If $\rho(M,N)\!\downarrow$, then $(M, \rho(M,N)) \in U$ and $\gamma(\rho(M,N)) = N$.

(upt:3) For every $M \in \mathsf{LDB}(\mathbf{D})$, $\rho(M, \gamma(M)) = M$. [Identity updates are reflected as identities.]

(upt:4) If $\rho(M,N)\!\downarrow$, then $\rho(\rho(M,N), \gamma(M)) = M$. [Every view update is globally reversible.]

(upt:5) If $\rho(M,N_1)\!\downarrow$ and $\rho(\rho(M,N_1),N_2)\!\downarrow$, then $\rho(M,N_2) = \rho(\rho(M,N_1),N_2)$. [View update reflection is transitive.]

(upt:6) If $\rho(M,N)\!\downarrow$ and $\gamma(M) \leq_{\mathbf{V}} N$, then $M \leq_{\mathbf{D}} \rho(M,N)$. [View update reflects order.]

(upt:7) If $\rho(M_1,N_1)\!\downarrow$ with $M_1 \leq_{\mathbf{D}} \rho(M_1,N_1)$, then for all $M_2 \in \mathsf{LDB}(\mathbf{D})$ with $M_1 \leq_{\mathbf{D}} M_2 \leq_{\mathbf{D}} \rho(M_1,N_1)$, there is an $N_2 \in \mathsf{LDB}(\mathbf{V})$ with $\rho(M_1,N_2) = M_2$ and $\rho(M_2, \gamma(\rho(M_1,N_1))) = \rho(M_1,N_1)$. [This condition is called *order completeness*.]

(upt:8) If $M_1, M_2 \in \mathsf{LDB}(\mathbf{D})$ with $M_1 \leq_{\mathbf{D}} M_2$, then for every $N_1, N_2 \in \mathsf{LDB}(\mathbf{V})$ for which $N_1 \leq_{\mathbf{V}} N_2$, $\rho(M_1,N_1)\!\downarrow$, and $\rho(M_2,N_2)\!\downarrow$, if must be the case that $\rho(M_1,N_1) \leq_{\mathbf{D}} \rho(M_2,N_2)$. [This condition is called *order reflection*.]

The *induced update family* on $\mathbf{D}$ is the smallest subset of U which will support the updates in T. It is denoted $\equiv_\rho$ and is given by $\{(M_1,M_2) \in \mathsf{LDB}(\mathbf{D}) \mid (\exists N \in \mathsf{LDB}(\mathbf{V}))(\rho(M_1,N) = M_2)\}$.

3.2 Discussion of the order-based conditions (upt:6)-(upt:8) Although it is first formalized in 4.1, it is already intuitive that a pair $(M_1,M_2) \in \mathsf{LDB}(\mathbf{D}) \times \mathsf{LDB}(\mathbf{D})$ represents an insertion if $M_1 \leq_{\mathbf{D}} M_2$, and a deletion if $M_2 \leq_{\mathbf{D}} M_1$.

Condition upt:6 of 3.1 above simply states that the reflection of an insertion in the view schema $\mathbf{V}$ is an insertion in the main schema $\mathbf{D}$. In view of the reversibility condition (upt:4), this also implies that the reflection of a deletion is a deletion.

Condition (upt:7), order completeness, states that if $M_1 \leq_{\mathbf{D}} M_2 \leq_{\mathbf{D}} M_3$ in the base schema $\mathbf{D}$, and there is a view update $(\gamma(M_1), \gamma(M_3))$ which is reflected as (M_1, M_3), then the view update $(\gamma(M_1), \gamma(M_2))$ is also allowed, and is reflected as (M_1, M_2). In view of the reversibility condition (upt:4), the view update $(\gamma(M_2), \gamma(M_3))$ is also allowed, and is reflected as (M_2, M_3). Roughly speaking, potential updates which are sandwiched between existing updates are also allowed.

Condition (upt:8) states that sequences of updates themselves are ordered. It is much more easily understood in the context of complements, and so a discussion is deferred to 3.11.

While these latter two conditions, (upt:7) and (upt:8), are somewhat technical, it will be seen in the following that they are exactly what is required to ensure that the update strategy defines an order-based complement, and conversely.

3.3 Notational convention For 3.4 through 3.7 below, $\mathbf{D}$ will be a database schema, $\Gamma = (\mathbf{V}, \gamma)$ will be a view of $\mathbf{D}$, U and T will be closed update families for $\mathbf{D}$ and $\mathbf{V}$, respectively, and ρ will be an update strategy for T with respect to U.

3.4 Proposition *$\equiv_\rho$ is a an order-compatible equivalence relation for $\mathbf{D}$.*

PROOF: The reflexivity, symmetry, and transitivity of $\equiv_\rho$ follow from conditions (upt:3), (upt:4), and (upt:5), respectively. Order compatibility follows from (upt:7). □

3.5 The complementary view for an update strategy The *ρ-complement* of Γ, denoted $\tilde{\Gamma}^\rho = (\tilde{\mathbf{V}}^\rho, \tilde{\gamma}^\rho)$, is defined to have $\mathsf{LDB}(\tilde{\mathbf{V}}^\rho) = \mathsf{LDB}(\mathbf{D})/\equiv_\rho$, with the morphism $\tilde{\gamma}^\rho : \mathbf{D} \to \tilde{\mathbf{V}}^\rho$ given by $M \mapsto [M]_{\equiv_\rho}$. The order $\leq_{\tilde{\mathbf{V}}^\rho}$ is just that which makes $\tilde{\gamma}^\rho$ an open surjection. More specifically, $[M_1]_{\equiv_\rho} \leq_{\tilde{\mathbf{V}}^\rho} [M_2]_{\equiv_\rho}$ iff there are $M_3 \in [M_1]_{\equiv_\rho}$, $M_4 \in [M_2]_{\equiv_\rho}$, with the property that $M_3 \leq_{\mathbf{D}} M_4$.

The *reflection of T along γ* is defined to be the relation $\mathsf{Refl}_\gamma(T) = \{(M_1, M_2) \in \mathsf{LDB}(\mathbf{D}) \times \mathsf{LDB}(\mathbf{D}) \mid (\gamma(M_1), \gamma(M_2)) \in T\}$.

3.6 Proposition *$\{\Gamma, \tilde{\Gamma}^\rho\}$ forms a meet-complementary pair, with meet $\Theta_{\mathsf{Refl}_\gamma(T)} = (\mathsf{LDB}(\mathbf{D})/\mathsf{Refl}_\gamma(T),\ \theta_{\mathsf{Refl}_\gamma(T)})$.*

PROOF: First, it will be shown that $\gamma \otimes \tilde{\gamma}^\rho : \mathbf{D} \to \mathbf{V}\ {}_{\gamma\otimes_{\tilde{\gamma}^\rho}}\tilde{\mathbf{V}}^\rho$ is a bijection. It suffices to establish that it is an injection, *i.e.*, that $\mathsf{Congr}(\Gamma) \cap \mathsf{Congr}(\tilde{\Gamma}^\rho) = \{(M, M) \mid M \in \mathsf{LDB}(\mathbf{D})\}$. Let $(M_1, M_2) \in \mathsf{LDB}(\mathbf{D}) \times \mathsf{LDB}(\mathbf{D})$. If $(M_1, M_2) \in \mathsf{Congr}(\tilde{\Gamma}^\rho)$, then $(\exists N \in \mathsf{LDB}(\mathbf{V}))(\rho(M_1, N) = M_2)$; in particular, $N = \gamma(M_2)$. Since $(M_1, M_2) \in \mathsf{Congr}(\Gamma)$ iff $\gamma(M_1) = \gamma(M_2)$, condition (upt:3) ensures that $(M_1, M_2) \in \mathsf{Congr}(\Gamma)$ iff $M_1 = M_2$.

Next, it will be shown that $\gamma \otimes \tilde{\gamma}^\rho$ is open. Let $M_1, M_2 \in \mathsf{LDB}(\mathbf{D})$ be such that $([M_1]_\Gamma, [M_1]_{\tilde{\Gamma}^\rho}) \leq_{\mathbf{V}\ {}_{\gamma\otimes_{\tilde{\gamma}^\rho}}\tilde{\mathbf{V}}^\rho} ([M_2]_\Gamma, [M_2]_{\tilde{\Gamma}^\rho})$; *i.e.*, $\gamma(M_1) \leq_{\mathbf{V}} \gamma(M_2)$ and $\tilde{\gamma}^\rho(M_1) \leq_{\tilde{\mathbf{V}}^\rho} \tilde{\gamma}^\rho(M_2)$. The latter inequality implies that there are $M_3 \in [M_1]_{\tilde{\Gamma}^\rho}$ and $M_4 \in [M_2]_{\tilde{\Gamma}^\rho}$ with the property that $M_3 \leq_{\mathbf{D}} M_4$. By the definition of $\mathsf{Congr}(\tilde{\Gamma}^\rho)$, there are $N_1, N_2 \in \mathsf{LDB}(\mathbf{V})$

with the property that $M_3 = \rho(M_1, N_1)$ and $M_4 = \rho(M_2, N_2)$, whence condition (upt:8) mandates that $M_1 \leq_{\mathbf{D}} M_2$, as required.

Finally, it will be established that $\mathsf{Congr}(\Gamma) \circ \mathsf{Congr}(\tilde{\Gamma}^\rho) = \mathsf{Congr}(\tilde{\Gamma}^\rho) \circ \mathsf{Congr}(\Gamma) = \mathsf{Refl}_\gamma(T)$. It is immediate that $\mathsf{Congr}(\Gamma) \subseteq \mathsf{Refl}_\gamma(T)$ and $\mathsf{Congr}(\tilde{\Gamma}^\rho) \subseteq \mathsf{Refl}_\gamma(T)$. To establish the converse, let $(M_1, M_2) \in \mathsf{Refl}_\gamma(T)$. Then there is a $N_1 \in \mathsf{LDB}(\mathbf{D})$ with $\gamma(M_1) = \gamma(N_1)$ and $\rho(M_2, \gamma(M_1)) = N_1$. In other words, $(M_1, N_1) \in \mathsf{Congr}(\Gamma)$ and $(N_1, M_2) \in \mathsf{Congr}(\tilde{\Gamma}^\rho)$. Thus, $\mathsf{Refl}_\gamma(T) \subseteq \mathsf{Congr}(\gamma) \circ \mathsf{Congr}(\tilde{\Gamma}^\rho)$, and so $\mathsf{Congr}(\Gamma) \circ \mathsf{Congr}(\tilde{\Gamma}^\rho) \subseteq \mathsf{Refl}_\gamma(T) \circ \mathsf{Refl}_\gamma(T) = \mathsf{Refl}_\gamma(T)$ *i.e.*, $\mathsf{Refl}_\gamma(T) = \mathsf{Congr}(\Gamma) \circ \mathsf{Congr}(\tilde{\Gamma}^\rho)$. Since $\mathsf{Refl}_\gamma(T)$ is an equivalence relation, condition (b) of 2.14 is satisfied, and so $\{\Gamma, \tilde{\Gamma}^\rho\}$ is a meet-complementary pair with meet $\Theta_{\mathsf{Refl}_\gamma(T)}$. □

The following theorem is the order-based analog of the characterization first reported by Bancilhon and Spyratos in their seminal paper [3, Thm. 7.3].

3.7 Theorem — constant complement representation of update *For every* $(N_1, N_2) \in T$ *and* $M \in \mathsf{LDB}(\mathbf{D})$ *with* $\gamma(M) = N_1$, $\rho(M, N_2) = (\gamma \otimes \tilde{\gamma}^\rho)^{-1}(N_2, [M]_{\tilde{\Gamma}^\rho})$.

PROOF: Follows from 3.6 and the definition of $\tilde{\Gamma}^\rho$. □

Bancilhon and Spyratos also present a result which associates an update strategy for a closed update family T of $\Gamma_1 = (\mathbf{V}_1, \gamma_1)$ to each complement view Γ_2 which admits a translation of T (Γ_2-*translatable* in their terminology) [3, Thm. 7.1]. However, their characterization does not provide conditions under which Γ_2 admits such a translation. The result 3.9 below makes this characterization precise in terms of the existence and character of the meet $\Gamma_1 \wedge \Gamma_2$.

3.8 The update strategy associated with a meet complement Let $\{\Gamma_1 = (\mathbf{V}_1, \gamma_1), \Gamma_2 = (\mathbf{V}_2, \gamma_2)\}$ be a meet-complementary pair of the schema $\mathbf{D}$.

(a) Define $\mathsf{UpdStr}\langle\Gamma_1, \Gamma_2\rangle : \mathsf{LDB}(\mathbf{D}) \times \mathsf{LDB}(\mathbf{V}_1) \to \mathsf{LDB}(\mathbf{D})$ by $(M, N) \mapsto (\gamma_1 \otimes \gamma_2)^{-1}(N, \gamma_2(M))$. $\mathsf{UpdStr}\langle\Gamma_1, \Gamma_2\rangle$ is called the *update strategy* for Γ_1 with respect to Γ_2.

(b) Define $\mathsf{UpdFam}\langle\Gamma_1, \Gamma_2\rangle = \{(N_1, N_2) \in \mathsf{LDB}(\mathbf{V}_1) \times \mathsf{LDB}(\mathbf{V}_1) \mid \lambda\langle\Gamma_1, \Gamma_1 \wedge \Gamma_2\rangle(N_1) = \lambda\langle\Gamma_1, \Gamma_1 \wedge \Gamma_2\rangle(N_2)\}$. $\mathsf{UpdFam}\langle\Gamma_1, \Gamma_2\rangle$ is called the *update family* induced by Γ_2 on Γ_1.

3.9 Theorem *Let* $\{\Gamma_1 = (\mathbf{V}_1, \gamma_1), \Gamma_2 = (\mathbf{V}_2, \gamma_2)\}$ *be a meet-complementary pair of the schema* $\mathbf{D}$. *Then* $\mathsf{UpdFam}\langle\Gamma_1, \Gamma_2\rangle$ *is a closed update family for* Γ_1, *and* $\mathsf{UpdStr}\langle\Gamma_1, \Gamma_2\rangle$ *is an update strategy for* $\mathsf{UpdFam}\langle\Gamma_1, \Gamma_2\rangle$ *with respect to* $\mathsf{LDB}(\mathbf{D}) \times \mathsf{LDB}(\mathbf{D})$.

PROOF: First of all, $\mathsf{UpdFam}\langle\Gamma_1, \Gamma_2\rangle$ must be an order-compatible equivalence relation, since it is the congruence of a view. Next, it is completely straightforward to verify that $\mathsf{UpdStr}\langle\Gamma_1, \Gamma_2\rangle$ satisfies conditions (up1:1)-(upt:6) of 3.1. To show (upt:7), let $M_1, M_2 \in \mathsf{LDB}(\mathbf{D})$, $N_1 \in \mathsf{LDB}(\mathbf{V}_1)$, be such that $\mathsf{UpdStr}\langle\Gamma_1, \Gamma_2\rangle(M_1, N_1)\downarrow$ with $M_1 \leq_{\mathbf{D}} M_2 \leq_{\mathbf{D}} \mathsf{UpdStr}\langle\Gamma_1, \Gamma_2\rangle(M_1, N_1)$. Then $(\gamma_1 \otimes \gamma_2)(M_1)$ is of the form $(\gamma_1(M_1), P)$ for some $P \in \mathsf{LDB}(\mathbf{V}_2)$, and $(\gamma_1 \otimes \gamma_2)(\mathsf{UpdStr}\langle\Gamma_1, \Gamma_2\rangle(M_1, N_1)) = (N_1, P)$. Thus, $(\gamma_1 \otimes \gamma_2)(M_2) = (\gamma_1(M_2), P)$, whence $(\gamma_1(M_1), \gamma_2(M_2))$, $(\gamma_1(M_2), N_2) \in \mathsf{UpdFam}\langle\Gamma_1, \Gamma_2\rangle$.

Finally, to show that (upt:8) holds, let $M_1, M_2 \in \mathsf{LDB}(\mathbf{D})$ with $\gamma_1(M_1) \leq_{\mathbf{V}} \gamma_2(M_2)$. Then $(\gamma_1 \otimes \gamma_2)^{-1}(M_1)$ is of the form $(\gamma_1(M_1), P_1)$ and $(\gamma_1 \otimes \gamma_2)^{-1}(M_2)$ is of the form $(\gamma_1(M_2), P_2)$ for some $P_1, P_2 \in \mathsf{LDB}(\mathbf{V}_1)$. Now if $(\exists N_1, N_2 \in \mathsf{LDB}(\mathbf{V}_1))$ $(\mathsf{UpdStr}\langle\Gamma_1, \Gamma_2\rangle(M_1, N_1) \leq_{\mathbf{D}} \mathsf{UpdStr}\langle\Gamma_1, \Gamma_2\rangle(M_2, N_2))$, then $(\gamma_1 \otimes \gamma_2)^{-1}(\mathsf{UpdStr}\langle\Gamma_1, \Gamma_2\rangle(M_1, N_1)) = (Q_1, P_1)$ and $(\gamma_1 \otimes \gamma_2)^{-1}(\mathsf{UpdStr}\langle\Gamma_1, \Gamma_2\rangle(M_2, N_2)) = (Q_2, P_2)$, with $Q_1, Q_2 \in \mathsf{LDB}(\mathbf{V}_1)$ and $(Q_1, P_1) \leq_{\mathbf{V}_1 \,_{\gamma_1}\!\otimes_{\gamma_2} \mathbf{V}_2} (Q_2, P_2)$. In particular, $P_1 \leq_{\mathbf{V}_2} P_2$, whence $(\gamma_1(M_1), P_1) \leq_{\mathbf{V}_1 \,_{\gamma_1}\!\otimes_{\gamma_2} \mathbf{V}_2} (\gamma_1(M_2), P_2)$, and so $M_1 \leq_{\mathbf{D}} M_2$. $\square$

Note carefully the definition of $\mathsf{UpdFam}\langle\Gamma_1, \Gamma_2\rangle$ in the above. The allowable updates to Γ_1 under constant complement Γ_2 are precisely those which hold the meet $\Gamma_1 \wedge \Gamma_2$ constant. Thus, only $\Gamma_1 \wedge \Gamma_2$,and no further properties of Γ_2, influence which updates are allowed. However, it may very well influence how those updates are reflected, as illustrated by the Π_B-complement views Π_{BC} and Π'_{BC} to Π_{AB} in the example schema $\mathbf{E}_1$ in 1.2 and 1.3. This uniqueness issue will be addressed in the next section.

To close, the following corollary, which identifies explicitly the natural association between meet complements and update strategies, is presented. It follows from the combination of 3.7 and 3.9.

3.10 Corollary *Let $\mathbf{D}$ be a database schema, and let Γ be a view of $\mathbf{D}$. There is natural bijective correspondence between update strategies for the view Γ and (equivalence classes of) meet complements of that view. Specifically:*

(a) *For any update strategy ρ, $\mathsf{UpdStr}\langle\Gamma, \tilde{\Gamma}^\rho\rangle = \rho$.*

(b) *For any meet complement Γ_1 of Γ, $\tilde{\Gamma}^{\mathsf{UpdStr}\langle\Gamma,\Gamma_1\rangle} = \Gamma_1$.* $\square$

3.11 Discussion of condition (upt:8) It is now easy to see the rôle of condition (upt:8) of 3.1. If $M_1 \leq_{\mathbf{D}} M_2$ in $\mathbf{D}$; then $\gamma(M_1) \leq_{\mathbf{V}} \gamma(M_2)$ and $\tilde{\gamma}^\rho(M_1) \leq_{\tilde{\mathbf{V}}^\rho} \tilde{\gamma}^\rho(M_2)$. As M_1 and M_2 updated, the $\tilde{\mathbf{V}}^\rho$ component of this representation does not change; it will always be $\tilde{\gamma}^\rho(M_1)$ for M_1 and $\tilde{\gamma}^\rho(M_2)$ for M_2. Thus, whenever M_1 and M_2 are updated to states M'_1 and M'_2, respectively, for which $\gamma(M'_1) \leq_{\mathbf{V}} \gamma(M'_2)$, it must be that $(\gamma(M'_1), \tilde{\gamma}^\rho(M_1)) \leq_{\mathbf{V} \,_{\gamma}\!\otimes_{\tilde{\gamma}^\rho} \tilde{\mathbf{V}}^\rho} (\gamma(M'_2), \tilde{\gamma}^\rho(M_2))$, whence $M'_1 \leq_{\mathbf{D}} M'_2$.

4. Uniqueness and Merging of Update Strategies

The necessary background having been established, the main results on the uniqueness of view update strategies and meet complements in the order-based context are presented. First, it is necessary to make precise the relationship between the order properties and types of updates.

4.1 Types of updates Let $\mathbf{D}$ be a database schema, and let U be a closed update family for $\mathbf{D}$.

(a) A pair $(M_1, M_2) \in U$ is called:

(i) a *formal insertion* with respect to U if $M_1 \leq_{\mathbf{D}} M_2$;

(ii) a *formal deletion* with respect to U if $M_2 \leq_{\mathbf{D}} M_1$;

(iii) an *order-based update* with respect to U if there exists a nonempty sequence $(N_1, N_2), (N_2, N_3), \ldots (N_{k-2}, N_{k-1}), (N_{k-1}, N_k)$ of elements of U with the properties that $N_1 = M_1$, $N_k = M_2$, and each pair (N_i, N_{i+1}), $1 \leq i \leq k-1$, is either a formal insertion or else a formal deletion with respect to U.

(b) The update family U is called *order realizable* if every pair in U is an order-based update.

A central result of this paper states that within the constant-complement order-based framework, the reflection of an order-based update of a view to the base schema is unique, without qualification. It does not depend upon the choice of complement, and it does not matter whether other updates are or are not order based. In short, the reflection of order-based updates is unique in a strong and global sense. The proof offered in 4.2 and 4.3 below is a substantial simplification over that given in the earlier conference version [16].

4.2 Lemma — characterization of formal insertions and deletions *Let* $\mathbf{D}$ *be a database schema, let* $\Gamma = (\mathbf{V}, \gamma)$ *be a view of* $\mathbf{D}$*, and let* U *and* T *be closed update families for* $\mathbf{D}$ *and* $\mathbf{V}$*, respectively. Let* ρ *be an update strategy for* T *with respect to* U*, and let* $M \in \mathsf{LDB}(\mathbf{D})$ *and* $N \in \mathsf{LDB}(\mathbf{V})$ *be such that* $\rho(M, N)\downarrow$*. Then the following hold.*

(a) *If* $(\gamma(M), N)$ *is a formal insertion; i.e., if* $\gamma(M) \leq_{\mathbf{V}} N$*, then* $\rho(M, N) = \inf(\{P \in \gamma^{-1}(N) \mid M \leq_{\mathbf{D}} P\})$.

(b) *Dually, if* $(\gamma(M), N)$ *is a formal deletion; i.e., if* $N \leq_{\mathbf{V}} \gamma(M)$*, then* $\rho(M, N) = \sup(\{P \in \gamma^{-1}(N) \mid P \leq_{\mathbf{D}} M\})$.

Proof: To show (a), first note that any $P \in \gamma^{-1}(N)$ is of the form $(\gamma \otimes \tilde{\gamma}^{\rho})^{-1}(N, \tilde{\gamma}^{\rho}(P))$. In particular, in view of 3.7, $\rho(M, N) = (\gamma \otimes \tilde{\gamma}^{\rho})^{-1}(N, \tilde{\gamma}^{\rho}(M))$. Now if $M \leq_{\mathbf{D}} P$, then $\rho(M, N) = (\gamma \otimes \tilde{\gamma}^{\rho})^{-1}(N, \tilde{\gamma}^{\rho}(M)) \leq_{\mathbf{D}} (\gamma \otimes \tilde{\gamma}^{\rho})^{-1}(N, \tilde{\gamma}^{\rho}(P)) = P$. Thus, $\rho(M, N) = \inf(\{P \in \gamma^{-1}(N) \mid M \leq_{\mathbf{D}} P\})$, as claimed. The proof of (b) is dual. □

The remarkable property of the above characterization is that, for formal insertions and formal deletions, the formulas for $\rho(M, N)$ depend only upon the view Γ, and not upon the specific choice of complement. This means that a given update is always reflected in the same way, regardless of the complement. This is recaptured formally in the theorem below.

4.3 Theorem — uniqueness of reflection of order-based view updates *Let* $\mathbf{D}$ *be a database schema, let* $\Gamma = (\mathbf{V}, \gamma)$ *be a view of* $\mathbf{D}$*, and let* U *and* T *be a closed update families for* $\mathbf{D}$ *and* $\mathbf{V}$*, respectively. Let* ρ_1 *and* ρ_2 *be update strategies for* T *with respect to* U*. Then, for any* $M \in \mathsf{LDB}(\mathbf{D})$ *and* $N \in \mathsf{LDB}(\mathbf{V})$ *with* $(\gamma(M), N) \in T$ *an order-based update, it must be the case that* $\rho_1(M, N) = \rho_2(M, N)$*. In particular, if* T *is order realizable, then* $\rho_1 = \rho_2$*.* □

Although it is primarily uniqueness of view update strategies which is the focus of this work, it is nonetheless worthwhile to note that in the case that the update family in the view is order realizable, the view complement defined by that update family is unique.

4.4 Corollary — uniqueness of meet complements *Let* $\mathbf{D}$ *be a database schema, let* $\Gamma = (\mathbf{V}, \gamma)$ *be a view of* $\mathbf{D}$*, and* T *be an order-realizable closed update family for* $\mathbf{V}$*. Then, up to isomorphism, there is a unique view of* $\mathbf{D}$ *which is a* Θ_T*-complement of* Γ*.*

PROOF: Follows immediately from 3.10 and the above theorem. □

4.5 Example — Illustration of uniqueness Recall the schema $\mathbf{E}_1$ from 1.2, with relation schema $R[ABC]$, constrained by the functional dependency $B \to C$. The view to be updated is Π_{AB}. From 2.17, it follows that Π_{BC} is a Π_B-complement of Π_{AB}, and the corresponding update family on Π_{AB} is precisely that which holds Π_B constant. This update family is order realizable, since there are no constraints on Π_{AB}. Any change can be realized by first inserting the new tuples, and then deleting the ones which are no longer wanted. Thus, in view of 4.3, the only update strategy in the sense of 3.1 which supports these updates is that which holds Π_{BC} constant. Furthermore, 4.4 above guarantees that Π_{BC} is the only order-based Π_B-complement of Π_{AB}. In particular, in view of 2.5, Π_{BC} is the only SPJR-view of $\mathbf{E}_1$ which is a Π_B-complement of Π_{AB}. In short, the theory has very strong things to say about $\mathbf{E}_1$.

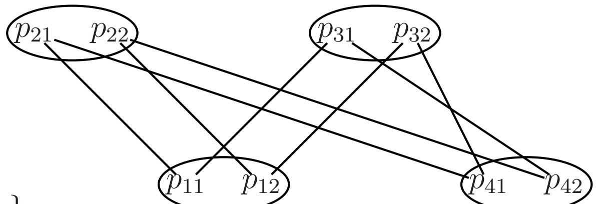
4.6 Example — using tuple ordering to force order realizability Now, revisit the example schema $\mathbf{E}_2$ from 1.3. It is identical to $\mathbf{E}_1$, save that it is constrained by the additional FD $B \to A$. Π_{BC} is still a Π_B-complement of Π_{AB}, since the dependency $B \to A$ embeds in Π_{AB}. Thus, the update family in which Π_B is held constant is again that induced on Π_{AB} by the complement Π_{BC}. However, this time the states in Π_{AB} are constrained by the FD $B \to A$. This has the consequence that none of the allowable updates on Π_{AB} is order based; thus, under the natural order, the theory developed above has nothing to say either about updates to or about Π_B-complements of Π_{AB}.

Fortunately, in this situation, there is a way to establish results similar to those for $\mathbf{E}_1$. The trick is to add additional order to the relational states. Let $\preceq_A$, $\preceq_B$ and $\preceq_C$ be partial orders on $\mathsf{Dom}(A)$, $\mathsf{Dom}(B)$ and $\mathsf{Dom}(C)$, respectively. Define the ordering $\preceq_{ABC}$ on tuples over ABC by $(a_0, b_0, c_0) \preceq_{ABC} (a_1, b_1, c_1)$ iff $a_0 \preceq_A a_1$, $b_0 \preceq_B b_1$, and $c_0 \preceq_C c_1$. Finally, extend the ordering $\preceq_{ABC}$ to relations on ABC by $M_1 \preceq_{ABC} M_2$ iff $(\forall t_0 \in M_1)(\exists t_1 \in M_2)(t_0 \preceq_{ABC} t_1)$. It is easy to see that this induces a partial order on the relations of $R[ABC]$ which is stronger than $\leq_{\mathbf{E}_2}$ in general, and the same as $\leq_{\mathbf{E}_2}$ if each of the three orders $\preceq_A$, $\preceq_B$, and $\preceq_C$ is flat (*i.e.*, no ordering at all). Define a similar ordering, using the same base orders $\preceq_A$, $\preceq_B$, and $\preceq_C$, on the states of $R[AB]$ and $R[BC]$.

Now, for the specific problem at hand, let $\preceq_A$ be any *total* order on $\mathsf{Dom}(A)$, and let $\preceq_B$ and $\preceq_C$ be the flat orders on $\mathsf{Dom}(B)$ and $\mathsf{Dom}(C)$, respectively. Using the orders $\preceq_{ABC}$ on $R[ABC]$, $\preceq_{AB}$ on $R[AB]$, and $\preceq_{BC}$ on $R[BC]$, every allowable update to Π_{AB} under constant complement Π_{BC} becomes order realizable. Furthermore, Π_{BC} is completely unchanged. Thus, the above results apply; update with constant complement Π_{BC} is the unique order-based update strategy on Π_{AB} with constant complement Π_B, and Π_{BC} is the unique order-based Π_B-complement of Π_{AB}.

Given that two update strategies ρ_1 for T_1 and ρ_2 for T_2 on the same view of a schema $\mathbf{D}$ must agree on view updates in $T_1 \cap T_2$, it is natural to ask whether ρ_1 and ρ_2 may be combined into an update strategy for the smallest set of updates containing $T_1 \cup T_2$. The next two examples demonstrate that this is unfortunately not the case.

4.7 Example — Non-mergeable update strategies Let $\mathbf{D}$ have $\mathsf{LDB}(\mathbf{D}) = \{p_{11}, p_{12}, p_{21}, p_{22}, p_{31}, p_{32}, p_{41}, p_{42}\}$ with the Hasse diagram ([6, 1.15]) shown to the right. The ellipses identify the states which are in the same equivalence class of the view $\Gamma = (\mathbf{V}, \gamma)$. Thus, $\{p_{11}, p_{12}\} \leq_{\mathbf{V}} \{p_{21}, p_{22}\}$, $\{p_{11}, p_{12}\} \leq_{\mathbf{V}} \{p_{31}, p_{32}\}$, $\{p_{41}, p_{42}\} \leq_{\mathbf{V}} \{p_{21}, p_{22}\}$, and $\{p_{41}, p_{42}\} \leq_{\mathbf{V}} \{p_{31}, p_{32}\}$. Let T_1 be the smallest equivalence relation containing $(\{p_{11}, p_{12}\}, \{p_{21}, p_{22}\})$ and $(\{\{p_{11}, p_{12}\}, \{p_{31}, p_{32}\}\})$, and let T_2 be the smallest equivalence relation containing $(\{p_{41}, p_{42}\}, \{p_{21}, p_{22}\})$ and $(\{p_{41}, p_{42}\}, \{p_{31}, p_{32}\})$.



It is easy to see that T_1 and T_2 are order realizable, and that there are unique update strategies ρ_1 and ρ_2 for T_1 and T_2, respectively. For example, $\rho_1(p_{11}, \{p_{31}, p_{32}\}) = p_{31}$, and $\rho_2(p_{41}, \{p_{31}, p_{32}\}) = p_{32}$. However, there can be no update strategy for the smallest set of updates containing $T_1 \cup T_2$. Indeed, if ρ were such an update strategy, then by following the path $\{p_{11}, p_{12}\} \to \{p_{21}, p_{22}\} \to \{p_{41}, p_{42}\}$, it must be that $\rho(p_{11}, \{p_{41}, p_{42}\}) = p_{41}$. However, by following the path $\{p_{11}, p_{12}\} \to \{p_{31}, p_{32}\} \to \{p_{41}, p_{42}\}$, it must be that $\rho(p_{11}, \{p_{41}, p_{42}\}) = p_{42}$. Visually, one can see that the problem is that there is a cycle $(p_{11}, p_{21}, p_{41}, p_{32}, p_{12})$ in the Hasse diagram of $\mathsf{LDB}(\mathbf{D})$ which connects two distinct points in the same equivalence class of the $\mathsf{Congr}(\Gamma)$. These ideas will be formalized more completely in the next section.

4.8 Example — Further non-mergeable update strategies The previous example is somewhat abstract and may be viewed as contrived; and it is therefore natural to ask whether the same problem can occur with "realistic" examples. The answer is, unfortunately, once again yes, as the following example illustrates. Let the base schema $\mathbf{E}_3$ have the single relation $R[ABC]$, governed by the functional dependencies $A \to C$ and $B \to C$. Let the view to be updated be $\Pi_{AB} = (R[AB], \pi_{AB})$, the projection of the main relation onto the attributes AB. Define the update family T_1 on $\mathsf{LDB}(R[AB])$ to be the smallest family of order-based updates which contains the following:

(i) The insertion of (a, b) is allowed into state $N \in \mathsf{LDB}(R[AB])$ iff $a \notin \pi_A(N)$ and $b \in \pi_B(N)$.

(ii) The deletion of (a, b) is allowed from state $N \in \mathsf{LDB}(R[AB])$ iff $a \notin \pi_A(N \setminus \{(a, b)\})$ and $b \in \pi_B(N \setminus \{(a, b)\})$.

There is a simple way to view this family of updates. Perform a horizontal decomposition of the states N of $R[AB]$ relative to the functional dependency $A \to B$, in the sense of [7]. One component contains the largest subset N_2 of N which satisfy the *afunctional dependency* $A \not\twoheadrightarrow B$, in the sense that for every $(a, b) \in N_2$, there is another tuple $(a, b') \in N_2$ with $b \neq b'$. The other component $N_1 = N \setminus N_2$ thus satisfies the functional dependency $A \to B$. Updates are allowed only to N_1, and only in such a way that $\pi_B[N]$ is held constant and $A \to B$ is preserved. Both N_2 and $\pi_B(N)$ are held constant during all updates. It is easy to see that there is a unique update strategy ρ_1 for T_1. The functional dependency

$B \to C$ ensures that the C entry may be recovered uniquely.

Similarly, the family T_2 of updates is defined on $\mathsf{LDB}(R[AB])$ by reversing the rôles of A and B. In T_2, the relation is decomposed into a component satisfying $B \to A$ and a component satisfying $B \not\twoheadrightarrow A$, and updates are allowed only to the component satisfying $B \to A$. The corresponding update strategy is denoted ρ_2. In this case, it is $A \to C$ which ensures that the C entry may be recovered uniquely.

Now let $M = \{(a_0, b_0, c_0), (a_1, b_1, c_1)\}$. The diagram of Fig. 2 shows two update sequences, each resulting in a final state of $\{(a_0, b_1), (a_2, b_2)\}$ of the view. In this diagram, $\mathsf{I}_1(-)$ denotes an insertion in ρ_1, $\mathsf{D}_2(-)$ a deletion in ρ_2, and so forth. Although the

$$
\begin{tikzcd}[column sep=small]
\begin{smallmatrix} a_0 & b_0 & c_0 \\ a_1 & b_1 & c_1 \\ a_2 & b_1 & c_1 \end{smallmatrix} \arrow[r, "\mathsf{D}_1(a_1,b_1)"] & \begin{smallmatrix} a_0 & b_0 & c_0 \\ a_2 & b_1 & c_1 \end{smallmatrix} \arrow[r, "\mathsf{I}_2(a_2,b_2)"] & \begin{smallmatrix} a_0 & b_0 & c_0 \\ a_2 & b_1 & c_1 \\ a_2 & b_2 & c_1 \end{smallmatrix} \arrow[r, "\mathsf{D}_2(a_2,b_1)"] & \begin{smallmatrix} a_0 & b_0 & c_0 \\ a_2 & b_2 & c_1 \end{smallmatrix} \arrow[r, "\mathsf{I}_2(a_0,b_1)"] & \begin{smallmatrix} a_0 & b_0 & c_0 \\ a_0 & b_1 & c_0 \\ a_2 & b_2 & c_1 \end{smallmatrix} \arrow[r, "\mathsf{D}_2(a_0,b_0)"] & \begin{smallmatrix} a_0 & b_1 & c_0 \\ a_2 & b_2 & c_1 \end{smallmatrix} \\
\begin{smallmatrix} a_0 & b_0 & c_0 \\ a_1 & b_1 & c_1 \end{smallmatrix} \arrow[u, "\mathsf{I}_1(a_2,b_1)"'] \arrow[d, "\mathsf{I}_1(a_2,b_0)"] & & & & & \\
\begin{smallmatrix} a_0 & b_0 & c_0 \\ a_2 & b_0 & c_0 \\ a_1 & b_1 & c_1 \end{smallmatrix} \arrow[r, "\mathsf{D}_1(a_0,b_0)"] & \begin{smallmatrix} a_2 & b_0 & c_0 \\ a_1 & b_1 & c_1 \end{smallmatrix} \arrow[r, "\mathsf{I}_2(a_2,b_2)"] & \begin{smallmatrix} a_2 & b_0 & c_0 \\ a_2 & b_2 & c_0 \\ a_1 & b_1 & c_1 \end{smallmatrix} \arrow[r, "\mathsf{D}_2(a_2,b_0)"] & \begin{smallmatrix} a_2 & b_2 & c_0 \\ a_1 & b_1 & c_1 \end{smallmatrix} \arrow[r, "\mathsf{I}_1(a_0,b_1)"] & \begin{smallmatrix} a_1 & b_1 & c_1 \\ a_0 & b_1 & c_1 \\ a_2 & b_2 & c_0 \end{smallmatrix} \arrow[r, "\mathsf{D}_1(a_1,b_1)"] & \begin{smallmatrix} a_0 & b_1 & c_1 \\ a_2 & b_2 & c_0 \end{smallmatrix}
\end{tikzcd}
$$

Figure 2: Combining update strategies in an inconsistent fashion

final states have the same π_{AB} projection, they are not the same. Thus, these two update strategies cannot be combined into a common update strategy which subsumes the updates of each.

4.9 Mergeability and normalization Since the schema $\mathbf{E}_3$ of 4.8 is not even in second normal form (2NF), it is interesting to ask whether problems of non-mergeability can occur when further normalization is enforced. The answer is that the problem remains even in the presence of full normalization into Boyce-Codd normal form (BCNF) [1, 11.2.1]. Indeed, the set of three views $\{\Pi_{AB} = (R[AB], \pi_{AB}), \Pi_{AC} = (R[AC], \pi_{AC}), \Pi_{BC} = (R[BC], \pi_{BC})\}$ of the schema $\mathbf{E}_3$ is easily seen to form a lossless and dependency-preserving decomposition into BCNF. Nonetheless, the problems associated with updates to Π_{AB} remain. Even in the case that the foreign-key dependencies $\Pi_{AB}[A] \subseteq \Pi_{AC}[A]$ and $\Pi_{AB}[B] \subseteq \Pi_{BC}[B]$ are enforced, it is easy to construct the update anomaly which is illustrated in Fig. 2, provided that the initial states of Π_{AC} and Π_{BC} are chosen to be rich enough that the foreign-key dependencies are not violated during the updates. Note that a similar conclusion regarding normalization was reached with schema $\mathbf{C}_1$ of 1.1.

5. Direct Construction of Update Strategies

The results of 4.2 and 4.3 establish that the way in which an order-based update (N_1, N_2) of a view Γ is reflected back to the base schema $\mathbf{D}$ is intrinsic to the view, and not dependent upon the choice of update strategy (or, equivalently, upon the choice of complement). On the other hand, the examples of 4.7 and 4.8 show that all update pairs (N_1, N_2) which are supported in some update strategy cannot simply be incorporated into a single, all-encompassing update strategy. Nonetheless, this uniqueness result has substantial implications, and suggests that a study of the properties of the view Γ may reveal substantial information as to which sets of potential update pairs may be combined into a common update strategy. It is the focus of this section to initiate such an investigation.

The starting point is to axiomatize the essential properties of update pairs which were identified in 4.2.

5.1 Projective and strong order pairs Let $\mathbf{D}$ be a database schema, let $\Gamma = (\mathbf{V}, \gamma)$ be a view of $\mathbf{D}$, and let $(N_1, N_2) \in \mathsf{LDB}(\mathbf{V}) \times \mathsf{LDB}(\mathbf{V})$ with $N_1 \leq_{\mathbf{V}} N_2$.

(a) (N_1, N_2) is said to be *upward projective* if for each $M_1 \in \gamma^{-1}(N_1)$, $\inf(\{M_2 \in \gamma^{-1}(N_2) \mid M_1 \leq_{\mathbf{D}} M_2\})$ exists and lies in $\gamma^{-1}(N_2)$. In this case, $\inf(\{M_2 \in \gamma^{-1}(N_2) \mid M_1 \leq_{\mathbf{D}} M_2\})$ is denoted $\mathsf{Insert}_{\mathbf{V}}(M_1, N_2)$.

(b) (N_1, N_2) is said to be *downward projective* if for each $M_2 \in \gamma^{-1}(N_2)$, $\sup(\{M_1 \in \gamma^{-1}(N_1) \mid M_1 \leq_{\mathbf{D}} M_2\})$ exists and lies in $\gamma^{-1}(N_1)$. In this case, $\sup(\{M_1 \in \gamma^{-1}(N_1) \mid M_1 \leq_{\mathbf{D}} M_2\})$ denoted $\mathsf{Delete}_{\mathbf{V}}(M_1, N_2)$.

(c) (N_1, N_2) is said to be a *strong order pair* if it is both upward and downward projective, and, in addition, the association $M_1 \mapsto \mathsf{Insert}_{\mathbf{V}}(M_1, N_2)$ is an order isomorphism of $\gamma(N_1)$ and $\gamma(N_2)$ (under the ordering inherited from $\leq_{\mathbf{D}}$) with inverse $M_2 \mapsto \mathsf{Delete}_{\mathbf{V}}(M_2, N_2)$. Define $\mathsf{SPairs}(\Gamma)$ to be the set of all strong order pairs of Γ.

5.2 Notational convention Unless stated specifically to the contrary, throughout this section, let $\mathbf{D}$ be a database schema, let $\Gamma = (\mathbf{V}, \gamma)$ be a view of $\mathbf{D}$, and let $\mathfrak{P}$ be a reflexive subset of $\mathsf{SPairs}(\Gamma)$. Furthermore, let $U = \mathsf{LDB}(\mathbf{D}) \times \mathsf{LDB}(\mathbf{D})$.

Before proceeding further, it is important to characterize the fundamental upper and lower bounds on a closed update family for the view Γ. As is shown below, any single strong order pair defines a closed update family for which there exists an update strategy, and the set of all strong order pairs provides a strict upper bound.

5.3 The update family of $\mathfrak{P}$ and universal update families

(a) Define $\mathfrak{P}^R = \{(N_2, N_1) \mid (N_1, N_2) \in \mathfrak{P}\}$.

(b) The *order-based update family* defined by $\mathfrak{P}$ is the smallest equivalence relation containing $\mathfrak{P}$, and is denoted $\mathsf{OrdUpd}(\mathfrak{P})$. Note that $\mathsf{OrdUpd}(\mathfrak{P})$ is just the closure under relational composition of $\mathfrak{P} \cup \mathfrak{P}^R$.

(c) $\mathfrak{P}$ is *universal* for Γ if $\mathsf{OrdUpd}(\mathfrak{P})$ is the largest closed update family for $\mathsf{LDB}(\mathbf{V})$ for which there exists an update strategy with respect to U. In other words, $\mathfrak{P}$ is universal if whenever ρ is an order-based update strategy for T with respect to U, then $T \subseteq \mathsf{OrdUpd}(\mathfrak{P})$. In this case, $\mathsf{OrdUpd}(\mathfrak{P})$ is called the *universal update family* for Γ.

5.4 Proposition — bound on update strategies

(a) *For any* $(N_1, N_2) \in \mathsf{SPairs}(\Gamma)$, *there is an order-based update strategy* ρ *for* $T = \{(N_1, N_1), (N_2, N_2)(N_1, N_2), (N_2, N_1)\}$ *with respect to* U.

(b) *If* $\mathfrak{P}$ *is universal for* Γ, *then it is all of* $\mathsf{SPairs}(\Gamma)$.

PROOF: For part (a), define ρ by $(M_1, N_1) \mapsto M_1$ and $(M_1, N_2) \mapsto \mathsf{Insert}_{\mathbf{V}}(M_1, N_2)$ for $M_1 \in \gamma^{-1}(N_1)$, $(M_2, N_1) \mapsto M_2$ and $(M_2, N_2) \mapsto \mathsf{Delete}_{\mathbf{V}}(M_2, N_1)$ for $M_2 \in \gamma^{-1}(N_2)$, and undefined otherwise. It is trivial to verify that ρ satisfies the conditions of 3.1. Part (b) now follows directly from (a) and 4.2. $\square$

Although simple, part (b) of the above proposition is important, as it rules out the possibility of looking for subsets of $\mathsf{SPairs}(\Gamma)$ which cannot participate in any update strategy.

Next, some further basic definitions regarding updates within a view are developed.

5.5 Fundamental update-related notions

(a) For $M \in \mathsf{LDB}(\mathbf{D})$ and $N \in \mathsf{LDB}(\mathbf{V})$, define

$$\mathsf{PreUpdStr}\langle\Gamma, \mathfrak{P}\rangle(M, N) = \begin{cases} \mathsf{Insert}_{\mathbf{V}}(M, N) & \text{if } (\gamma(M), N) \in \mathfrak{P}. \\ \mathsf{Delete}_{\mathbf{V}}(M, N) & \text{if } (N, \gamma(M)) \in \mathfrak{P}. \\ \text{undefined} & \text{otherwise.} \end{cases}$$

$\mathsf{PreUpdStr}\langle\Gamma, \mathfrak{P}\rangle$ is the "skeleton" of updates from which any update strategy based upon $\mathfrak{P}$ is built.

(b) A *view update sequence* from $\mathfrak{P}$ is a nonempty sequence $\sigma = \langle N_1, N_2, \ldots, N_m\rangle$ of elements of $\mathfrak{P}$ with the property that $(N_i, N_{i+1}) \in \mathfrak{P} \cup \mathfrak{P}^R$ for each i, $1 \le i \le m-1$. The sequence σ is called *monotonically increasing* (resp. *monotonically decreasing*) if $N_i \le_{\mathbf{V}} N_{i+1}$ (resp. $N_{i+1} \le_{\mathbf{V}} N_i$) for $1 \le i \le m-1$. The term *monotonic* is used to describe a sequence which is either monotonically increasing or monotonically decreasing. Finally, σ is called *alternating* if $(N_i, N_{i+1}) \in \mathfrak{P}$ when i is odd and $(N_i, N_{i+1}) \in \mathfrak{P}^R$ when i is even. In the definition of alternating, there is no loss of generality in requiring the sequence to begin with an insertion (a pair from $\mathfrak{P}$) rather than a deletion (a pair from $\mathfrak{P}^R$), since it is always possible to prepend a dummy identity insertion (N_1, N_1).

(c) Let $\sigma = N_1, N_2, \ldots, N_m$ be a view update sequence from $\mathfrak{P}$, and let $M \in \gamma^{-1}(N_1)$. The associated *reflected update sequence* is $\mathsf{RefSeq}\langle\Gamma, \mathfrak{P}\rangle(\sigma, M) = \langle M_1, M_2, \ldots, M_m\rangle$, with $M_1 = M$, and $M_{i+1} = \mathsf{PreUpdStr}\langle\Gamma, \mathfrak{P}\rangle(M_i, N_i)$ for $1 \le i \le m-1$.

(d) Two view update sequences σ_1 and σ_2 from $\mathfrak{P}$ are *result equivalent* if $\mathsf{First}(\sigma_1) = \mathsf{First}(\sigma_2)$, and, for any $M \in \gamma^{-1}(\mathsf{First}(\sigma_1))$, $\mathsf{Last}(\mathsf{RefSeq}\langle\Gamma, \mathfrak{P}\rangle(\sigma_1, M)) = \mathsf{Last}(\mathsf{RefSeq}\langle\Gamma, \mathfrak{P}\rangle(\sigma_2, M))$.

(e) Define $\overline{\mathsf{PreUpdStr}}\langle\Gamma, \mathfrak{P}\rangle : \mathsf{LDB}(\mathbf{D}) \times \mathsf{LDB}(\mathbf{V}) \to \mathbf{2}^{\mathsf{LDB}(\mathbf{D})}$ by $M' \in \overline{\mathsf{PreUpdStr}}\langle\Gamma, \mathfrak{P}\rangle(M, N)$ iff there is a view update sequence σ from $\mathfrak{P}$ with $\mathsf{First}(\sigma) = \gamma(M)$, $\mathsf{Last}(\sigma) = N$, and $\mathsf{Last}(\mathsf{RefSeq}\langle\Gamma, \mathfrak{P}\rangle(\sigma, M)) = M'$. Note that $\overline{\mathsf{PreUpdStr}}\langle\Gamma, \mathfrak{P}\rangle$

is always a *total* function, although it is possible that $\overline{\mathsf{PreUpdStr}}\langle\Gamma, \mathfrak{P}\rangle(M, N) = \emptyset$. Of course, to define an update strategy, $\overline{\mathsf{PreUpdStr}}\langle\Gamma, \mathfrak{P}\rangle(M, N)$ must contain at most one element for each pair (M, N); this is axiomatized as *defining unique updates* in 5.8(c).

The following are simple but useful results based upon the preceding definitions.

5.6 Lemma *Let $\sigma = \langle N_1, N_2, \ldots, N_m\rangle$ be any view update sequence from $\mathfrak{P}$.*

(a) *The association $M \mapsto \mathsf{Last}(\mathsf{RefSeq}\langle\Gamma, \mathfrak{P}\rangle(\sigma, M))$ defines an isomorphism from $\gamma^{-1}(N_1)$ to $\gamma^{-1}(N_m)$, under the orderings on these sets which are inherited from $\mathsf{LDB}(\mathbf{D})$.*

(b) *For any $M \in \gamma^{-1}(N_1)$, $\mathsf{RefSeq}\langle\Gamma, \mathfrak{P}\rangle(\sigma \cdot \sigma^R, M) = M$.*

For (c) and (d) below, let σ be as above, and let σ_1 and σ_2 be additional view update sequences from $\mathfrak{P}$ with the properties that $\mathsf{First}(\sigma_1) = \mathsf{First}(\sigma_2)$ and $\mathsf{Last}(\sigma_1) = \mathsf{Last}(\sigma_2)$.

(c) *If $(\mathsf{Last}(\sigma_1), N_1) = (\mathsf{Last}(\sigma_2), N_1) \in \mathfrak{P}$, then $\sigma_1 \cdot \sigma$ and $\sigma_2 \cdot \sigma$ are result equivalent iff σ_1 and σ_2 are result equivalent.*

(d) *If $(N_m, \mathsf{First}(\sigma_1)) = (N_m, \mathsf{First}(\sigma_2)) \in \mathfrak{P}$, then $\sigma \cdot \sigma_1$ and $\sigma \cdot \sigma_2$ are result equivalent iff σ_1 and σ_2 are result equivalent.* □

5.7 Examples — basic properties related to constructing update strategies Fig. 3 presents a number of examples which illustrate some of the anomalies which must be overcome when attempting to construct an update strategy from a set of strong order pairs. In each, the points with labels of the form p_{ij} are states of the base schema $\mathbf{D}$, and the straight lines indicate the order relation using the standard conventions of Hasse diagrams [6, 1.15]. The ellipses surround points which map to the same equivalence class in the view Γ. Dashed lines are used to represent aspects of the order which are not recaptured in the strong order pairs of $\mathfrak{P}$.

Diagram (a) depicts a situation which violates the condition of *transitive completeness* defined in 5.8(b) below. Specifically, while the pair $(\{p_{11}, p_{12}\}, \{p_{31}, p_{32}\})$ is in the set $\mathfrak{P}$ of strong order pairs, the intermediate components $(\{p_{11}, p_{12}\}, \{p_{21}, p_{22}\})$ and $(\{p_{21}, p_{22}\}, \{p_{31}, p_{32}\})$ are not. With such an omission, it is generally impossible to construct an update strategy which satisfies the order completeness condition (upt:7) of 3.1.

Diagram (b) depicts a situation which violates the condition of *transitive closure* defined in 5.8(a) below. Here $(\{p_{11}, p_{12}\}, \{p_{21}, p_{22}\})$ and $(\{p_{21}, p_{22}\}, \{p_{31}, p_{32}\})$ are in $\mathfrak{P}$, but their composition $(\{p_{11}, p_{12}\}, \{p_{31}, p_{32}\})$ is not. This condition is fundamental, since without it monotonic updates cannot be composed. In virtually all of that which follows, transitive closure of $\mathfrak{P}$ will be mandated.

If the appropriate elements are included in $\mathfrak{P}$, then the situations depicted in (a) and (b) are well behaved, in the sense that all connections are strong order pairs, and there are no additional anomalies which hinder the construction of an update strategy. The remaining diagrams, (c) through (e), illustrate basic problems with the underlying base schema and view.

In diagram (c), the fundamental property that $(\{p_{11}, p_{12}\}, \{p_{31}, p_{32}\})$ be a strong order pair is violated. This shows that the composition of strong order pairs need not

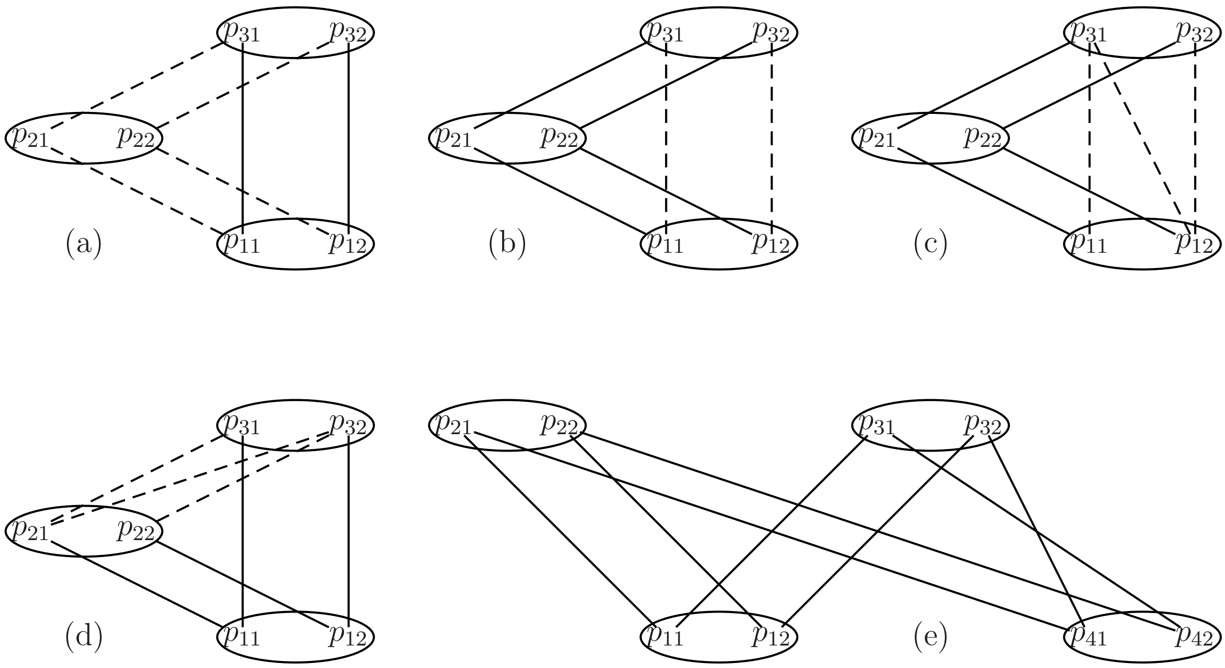


Figure 3: Some examples of views which fail certain conditions

be a strong order pair. Diagram (d) illustrates a situation in which transitive completeness is violated, and cannot be repaired by adding pairs to $\mathfrak{P}$. Diagram (e) illustrates a very fundamental problem, the lack of unique updates. Note that $\sigma_1 = \langle\{p_{11}, p_{12}\}, \{p_{21}, p_{22}\}, \{p_{41}, p_{42}\}\rangle$ is not result equivalent to $\sigma_2 = \langle\{p_{11}, p_{12}\}, \{p_{31}, p_{32}\}, \{p_{41}, p_{42}\}\rangle$, since $\mathsf{RefSeq}\langle\Gamma, \mathfrak{P}\rangle(\sigma_1, p_{11}) = p_{41}$, while $\mathsf{RefSeq}\langle\Gamma, \mathfrak{P}\rangle(\sigma_2, p_{11}) = p_{42}$. In the terminology of 5.8(c), $\mathfrak{P}$ does not *define unique updates*. Much of the work in this section is directed towards characterization of conditions which ensure such uniqueness.

5.8 Properties for ensuring unique update strategies

(a) $\mathfrak{P}$ is said to be *transitively closed* if for any view update sequence $\sigma = N_1, N_2, \ldots, N_m$, if $N_1 \leq_{\mathbf{V}} N_m$, then $(N_1, N_m) \in \mathfrak{P}$.

(b) $\mathfrak{P}$ is said to be *transitively complete* if for every $(N_1, N_2) \in \mathfrak{P}$ and $N_3 \in \mathsf{LDB}(\mathbf{V})$ with $N_1 \leq_{\mathbf{V}} N_3 \leq_{\mathbf{V}} N_2$, it is that case that (N_1, N_3), $(N_3, N_2) \in \mathfrak{P}$ as well.

(c) $\mathfrak{P}$ is said to *define unique updates* if for every $M, M_1, M_2 \in \mathsf{LDB}(\mathbf{D})$ and $N \in \mathsf{LDB}(\mathbf{V})$, if $M_1, M_2 \in \overline{\mathsf{PreUpdStr}}\langle\Gamma, \mathfrak{P}\rangle(M, N)$, then $M_1 = M_2$. In other words, for any $M \in \mathsf{LDB}(\mathbf{D})$ and $N \in \mathsf{LDB}(\mathbf{V})$, $\overline{\mathsf{PreUpdStr}}\langle\Gamma, \mathfrak{P}\rangle(M, N)$ is either a singleton or else empty.

(d) If $\mathfrak{P}$ defines unique updates, the associated *raw update strategy* $\mathsf{RawUpdStr}\langle\Gamma, \mathfrak{P}\rangle : \mathsf{LDB}(\mathbf{D}) \times \mathsf{LDB}(\mathbf{V}) \to \mathsf{LDB}(\mathbf{D})$ associated with $\mathfrak{P}$ is defined as follows.

$$\mathsf{RawUpdStr}\langle\Gamma, \mathfrak{P}\rangle(M, N) = \begin{cases} M' & \text{if } \overline{\mathsf{PreUpdStr}}\langle\Gamma, \mathfrak{P}\rangle(M, N) = \{M'\} \\ \text{undefined} & \text{if } \overline{\mathsf{PreUpdStr}}\langle\Gamma, \mathfrak{P}\rangle(M, N) = \emptyset \end{cases}$$

5.9 Observation *If $\mathfrak{P}$ is transitively closed, then it defines unique updates iff any two update sequences $\sigma_1 = \langle N_{11}, N_{12}, \ldots, N_{1m_1}\rangle$ and $\sigma_2 = \langle N_{21}, N_{22}, \ldots, N_{2m_2}\rangle$ with $N_{11} = N_{21}$ and $N_{1m_1} = N_{2m_2}$ are result equivalent.* $\square$

It is now possible to characterize those sets of strong order pairs which satisfy the first seven conditions of 3.1.

5.10 Theorem — partial characterization of update strategies *Let $\mathfrak{P}$ be transitively closed, and define $T = \mathsf{OrdUpd}(\mathfrak{P})$. Then $\rho = \mathsf{RawUpdStr}\langle\Gamma, \mathfrak{P}\rangle$ satisfies conditions (upt:1) through (upt:7) of 3.1 iff $\mathfrak{P}$ is transitively complete and defines unique updates.*

PROOF: First, assume that $\mathfrak{P}$ is transitively complete and defines unique updates. Conditions (upt:1) – (upt:4) are immediate from the definitions. Condition (upt:5) is a consequence of the requirement that $\mathfrak{P}$ define unique updates. (upt:6) follows from the definition of $\mathsf{PreUpdStr}\langle\Gamma, \mathfrak{P}\rangle$ and the fact that $\mathfrak{P}$ is transitively closed (so that all insertions and deletions are defined by $\mathsf{PreUpdStr}\langle\Gamma, \mathfrak{P}\rangle$). Finally, (upt:7) follows from the fact that $\mathfrak{P}$ is transitively complete.

Conversely, if $\rho = \mathsf{RawUpdStr}\langle\Gamma, \mathfrak{P}\rangle$ satisfies (upt:1) – (upt:7), then from (upt:5) it is immediate that $\mathfrak{P}$ defines unique updates, while transitive completeness follows from (upt:7). $\square$

Condition (upt:8) of 3.1, which requires that the view update strategy ρ possess the property of order reflection, remains to be addressed. Roughly speaking, it mandates that entire update sequences possess a form of ordering. In the view-based characterization of an update strategy, this property manifests itself in the requirement that the complement view $\tilde{\Gamma}^{\rho}$ be itself an order-based view, as shown in 3.11. In order to characterize it directly within the view Γ, it is necessary to introduce the concept of an orbit, and then two forms of ordering on orbits.

5.11 Orbits and their properties

(a) Let $M \in \mathsf{LDB}(\mathbf{D})$. The *orbit* of M with respect to $\langle\Gamma, \mathfrak{P}\rangle$ is $\mathsf{Orbit}\langle\Gamma, \mathfrak{P}\rangle(M) = \bigcup_{N \in \mathsf{LDB}(\mathbf{V})} \overline{\mathsf{PreUpdStr}}\langle\Gamma, \mathfrak{P}\rangle(M, N)$. In words, the orbit of M is just the set of all $M' \in \mathsf{LDB}(\mathbf{D})$ which can be reached from M via a sequence of updates from $\mathfrak{P} \cup \mathfrak{P}^R$. An *orbit* with respect to $\langle\Gamma, \mathfrak{P}\rangle$ is any set of this form.

Let μ_1 and μ_2 be orbits with respect to $\langle\Gamma, \mathfrak{P}\rangle$

(b) μ_1 and μ_2 are *parallel* if $\gamma(\mu_1) = \gamma(\mu_2)$, and *independent* if $\gamma(\mu_1) \cap \gamma(\mu_2) = \emptyset$.

(c) Write $\mu_1 \leq_{\langle\Gamma,\mathfrak{P}\rangle} \mu_2$ just in case $M_1 \leq_{\mathbf{D}} M_2$ for some $M_1 \in \mu_1$, $M_2 \in \mu_2$.

(d) Write $\mu_1 \trianglelefteq_{\langle\Gamma,\mathfrak{P}\rangle} \mu_2$ just in case $M_1 \leq_{\mathbf{D}} M_2$ for every $M_1 \in \mu_1$ and $M_2 \in \mu_2$ for which $\gamma(M_1) \leq_{\mathbf{V}} \gamma(M_2)$.

When the context is clear, particularly in examples, $\leq$ (resp. $\trianglelefteq$) will be written for the more cumbersome $\leq_{\langle\Gamma,\mathfrak{P}\rangle}$ (resp.$\trianglelefteq_{\langle\Gamma,\mathfrak{P}\rangle}$).

5.12 Examples — issues of orbits and order reflection Shown in Fig. 4 are some examples related to the condition of order reflection ((upt:8) of 3.1). The overall nota-

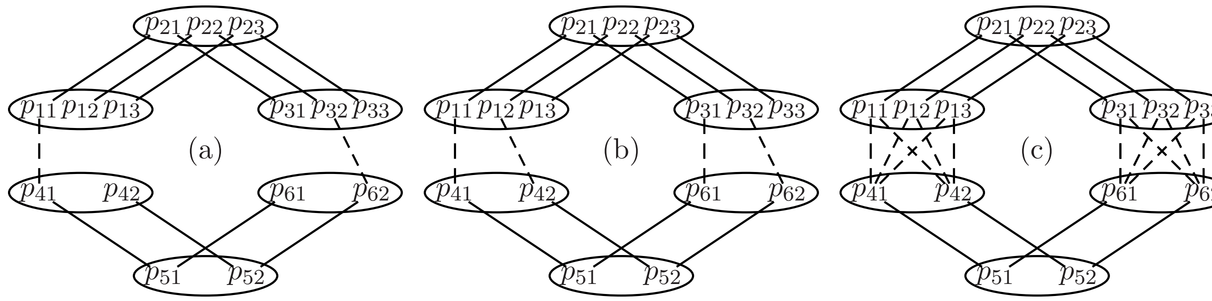


Figure 4: Examples related to orbits and order reflection

tional conventions are as in Fig. 3. In particular, the extant set of strong order pairs is $\mathfrak{P} = \{(\{p_{11}, p_{12}, p_{13}\}, \{p_{21}, p_{22}, p_{23}\}), (\{p_{11}, p_{12}, p_{13}\}, \{p_{31}, p_{32}, p_{33}\}), (\{p_{21}, p_{22}, p_{23}\}, \{p_{31}, p_{32}, p_{33}\}), (\{p_{41}, p_{42}\}, \{p_{51}, p_{52}\}), (\{p_{41}, p_{42}\}, \{p_{61}, p_{62}\}), (\{p_{51}, p_{52}\}, \{p_{61}, p_{62}\})\}$. There are five orbits, $\mu_1 = \{p_{11}, p_{21}, p_{31}\}$, $\mu_2 = \{p_{12}, p_{22}, p_{32}\}$, $\mu_3 = \{p_{13}, p_{23}, p_{33}\}$, $\mu_4 = \{p_{41}, p_{51}, p_{61}\}$, and $\mu_5 = \{p_{42}, p_{52}, p_{62}\}$, with the members of $\{\mu_1, \mu_2, \mu_3\}$ parallel, and the members of $\{\mu_4, \mu_5\}$ parallel. The others are independent.

In diagram (a), $\mu_4 \leq \mu_1$ and $\mu_5 \leq \mu_2$; for no orbits is it the case that $\mu_i \trianglelefteq \mu_j$. In diagram (b), $\mu_4 \trianglelefteq \mu_1$ and $\mu_5 \trianglelefteq \mu_2$. In diagram (c), $\mu_i \trianglelefteq \mu_j$ for $4 \leq i \leq 5$ and $1 \leq j \leq 3$. In (b) and (c), $\mathsf{RawUpdStr}\langle \Gamma, \mathfrak{P} \rangle$ satisfies the order reflection condition (upt:8), while in (a), it is not satisfied. The key property is that whenever $\mu_i \leq \mu_j$, then $\mu_i \trianglelefteq \mu_j$ as well. This is established formally in 5.15 below,

It should perhaps be remarked that it is quite possible that $\mu_i \leq \mu_j$ may hold for parallel orbits, although the examples shown here do not illustrate this. However, as shown in 5.13(b) below, it must then be the case that $\mu_i \trianglelefteq \mu_j$, so that ordering of parallel orbits has no direct impact on satisfaction of the order reflection condition.

5.13 Lemma *Assume that $\mathfrak{P}$ is transitively closed, and let μ_1 and μ_2 be orbits with respect to $\langle \Gamma, \mathfrak{P} \rangle$.*

(a) *μ_1 and μ_2 are either parallel or else independent. There are no other possibilities.*

(b) *If μ_1 and μ_2 are parallel, and $\mathfrak{P}$ furthermore defines unique updates, then $\mu_1 \leq_{\langle \Gamma, \mathfrak{P} \rangle} \mu_2$ implies $\mu_1 \trianglelefteq_{\langle \Gamma, \mathfrak{P} \rangle} \mu_2$.*

Proof: (a) Suppose that μ_1 and μ_2 are not independent. Then $\gamma(\mu_1) \cap \gamma(\mu_2) \neq \emptyset$, and so there exist $M_1 \in \mu_1$, $M_2 \in \mu_2$ such that $\gamma(M_1) = \gamma(M_2)$. If $M_1' \in \mu_1$ as well, then there is a view update sequence σ from $\mathfrak{P}$ with $\gamma(M_1) = \mathsf{First}(\sigma)$ and $M_1' = \mathsf{Last}(\mathsf{RefSeq}\langle \Gamma, \mathfrak{P} \rangle(\sigma, M_1))$. However, then $\mathsf{Last}(\mathsf{RefSeq}\langle \Gamma, \mathfrak{P} \rangle(\sigma, M_2))$ is also defined, with $\gamma(M_1') = \gamma(\mathsf{Last}(\mathsf{RefSeq}\langle \Gamma, \mathfrak{P} \rangle(\sigma, M_2)))$. Thus, $\gamma(\mu_1) \subseteq \gamma(\mu_2)$. Similarly, $\gamma(\mu_2) \subseteq \gamma(\mu_1)$, so that μ_1 and μ_2 are parallel.

(b) Suppose that μ_1 and μ_2 are parallel, that $\mathfrak{P}$ defines unique updates, and let $M_1 \in \mu_1$, $M_2 \in \mu_2$ with $M_1 \leq_{\mathbf{D}} M_2$. There are two possibilities to consider. First of all, if $\gamma(M_1) = \gamma(M_2)$, the result follows immediately from 5.6(a). If $\gamma(M_1) \neq \gamma(M_2)$, then since $M_1 \leq_{\mathbf{D}} M_2$, it must be that $\gamma(M_1) \leq_{\mathbf{V}} \gamma(M_2)$, and since μ_1 and μ_2 are parallel,

$(\gamma(M_1), \gamma(M_2)) \in \mathfrak{P}$. Thus $\mathsf{PreUpdStr}\langle\Gamma, \mathfrak{P}\rangle(M_1, \gamma(M_2)) = \mathsf{Insert}_{\mathbf{V}}(M_1, \gamma(M_2))$, whence $\mathsf{PreUpdStr}\langle\Gamma, \mathfrak{P}\rangle(M_1, \gamma(M_2)) \leq_{\mathbf{D}} M_2$, as required. □

5.14 The strong orbital linking condition $\mathfrak{P}$ satisfies the *strong orbital linking condition* if for any pair $\{\mu_1, \mu_2\}$ of independent orbits with respect to $\langle\Gamma, \mathfrak{P}\rangle$, $\mu_1 \leq_{\langle\Gamma,\mathfrak{P}\rangle} \mu_2$ implies $\mu_1 \trianglelefteq_{\langle\Gamma,\mathfrak{P}\rangle} \mu_2$.

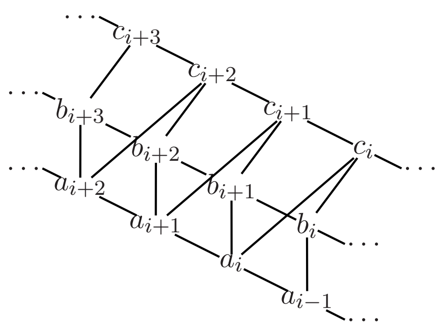
The strong orbital linking condition essentially recaptures condition (upt:8) of 3.1. If μ_1 is less than μ_2 somewhere (*i.e.*, $\mu_1 \leq_{\langle\Gamma,\mathfrak{P}\rangle} \mu_2$), then μ_1 is less than μ_2 in every case in which γ allows it (*i.e.*, $\mu_1 \trianglelefteq_{\langle\Gamma,\mathfrak{P}\rangle} \mu_2$). In view of 5.13, it is only necessary to consider independent orbits. This addition of the strong orbital linking condition is enough to extend 5.10 to a result which provides necessary and sufficient conditions for $\mathfrak{P}$ to define an update strategy.

5.15 Theorem — characterization of update strategies *Let* $\mathfrak{P} \subseteq \mathsf{SPairs}(\mathfrak{P})$ *be transitively closed, and let* $T = \mathsf{OrdUpd}(\mathfrak{P})$, $U = \mathsf{LDB}(\mathbf{D}) \times \mathsf{LDB}(\mathbf{D})$. *Then* $\mathsf{RawUpdStr}\langle\Gamma, \mathfrak{P}\rangle$ *is an update strategy (in the sense of 3.1) iff* $\mathfrak{P}$ *is transitively complete, defines unique updates, and satisfies the strong orbital linking condition.*

Proof: The combination of the strong orbital linking condition and 5.13(b) ensure that condition (upt:8) of 3.1 is satisfied. The rest of the proof follows immediately from 5.10. □

While interesting from a theoretical point of view, the conditions of 5.15 are often difficult to establish for examples which arise in practice. Therefore, it is useful to seek alternate, more useful conditions, which are sufficient but not strictly necessary for the construction of an update strategy from a view. The first task is to characterize a stronger version of transitive closure, called strong transitivity, which is satisfied by virtually all realistic examples.

5.16 Strong transitivity and the descending chain condition for views Recall that the definition of transitive closure of 5.8(a) states that for any view update sequence $\sigma = N_1, N_2, \ldots, N_m$, if $N_1 \leq_{\mathbf{V}} N_m$, then $(N_1, N_m) \in \mathfrak{P}$. However, this is a strictly weaker condition than requiring that σ and $\langle N_1, N_m\rangle$ be result equivalent, as the following example illustrates.



Consider the situation depicted in the diagram to the right. The states of the main schema $\mathbf{E}_{abc}$ consist of three subgroups; $A = \{a_i \mid i \in \mathbb{Z}\}$; $B = \{b_i \mid i \in \mathbb{Z}\}$; $C = \{c_i \mid i \in \mathbb{Z}\}$. Thus, $\mathsf{LDB}(\mathbf{E}_{abc}) = A \cup B \cup C$. Within each group, the set of elements is order isomorphic to the integers; thus $a_i \leq_{\mathbf{D}} a_j$, $b_i \leq_{\mathbf{D}} b_j$, $c_i \leq_{\mathbf{D}} c_j$ iff $i \leq j$. Furthermore, $a_i \leq_{\mathbf{D}} b_j$ iff $i \leq j - 1$, $a_i \leq_{\mathbf{D}} c_j$ iff $i \leq j$, and $b_i \leq_{\mathbf{D}} c_j$ iff $i \leq j$. The equivalence classes of the view $\Gamma_{abc} = (\mathbf{V}_{abc}, \gamma_{abc})$ are just A, B, and C. It is easy to see that $\mathfrak{P}_{abc} = \{(A, B), (B, C), (A, C)\}$ consists of strong order pairs. Yet, $\langle A, B, C\rangle$ and $\langle A, C\rangle$

are not result equivalent. For example, $\mathsf{RefSeq}\langle\Gamma_{[abc]}, \mathfrak{P}_{abc}\rangle(\langle A, B, C\rangle, a_i) = c_{i+1}$, yet $\mathsf{RefSeq}\langle\Gamma_{[abc]}, \mathfrak{P}_{abc}\rangle(\langle A, C\rangle, a_i) = c_i$.

In light of this, the following definition is appropriate.

(a) $\mathfrak{P}$ is *strongly transitive* if it is transitively closed and, for any $(N_1, N_2), (N_2, N_3) \in \mathfrak{P}$ and any $M \in \gamma^{-1}(N_1)$, $\mathsf{RefSeq}\langle\Gamma, \mathfrak{P}\rangle(\langle N_1, N_2, N_3\rangle, M) = \mathsf{RefSeq}\langle\Gamma, \mathfrak{P}\rangle(\langle N_1, N_3\rangle, M)$. In other words, $\mathfrak{P}$ is transitively closed if updating directly from N_1 to N_3 yields the same result as updating from N_1 to N_2 to N_3.

The above example illustrates a situation in which $\mathfrak{P}$ is transitively closed but not strongly transitive. This example is rather contrived, and, indeed, it requires quite an unusual schema to achieve this anomalous behavior. There is a natural condition which rules out this sort of behavior; specifically, the following definitions are appropriate.

(b) A subset $S \in \mathsf{LDB}(\mathbf{D})$ satisfies the *descending chain condition* [6, 2.37] if, given any left-infinite sequence $\ldots M_3 \leq_{\mathbf{D}} M_2 \leq_{\mathbf{D}} M_1 \leq_{\mathbf{D}} M_0$ of elements from S, there is an $n \in \mathbb{N}$ for which $M_{n+k} = M_n$ for all $k \in \mathbb{N}$.

(c) The view Γ satisfies the *local descending chain condition* if, for any $M \in \mathsf{LDB}(\mathbf{D})$, $\gamma(M)$ satisfies the descending chain condition under the order inherited from $\mathsf{LDB}(\mathbf{D})$.

(d) A database schema $\mathbf{D}$ satisfies the *finite submodels condition* if for any $M \in \mathsf{LDB}(\mathbf{D})$, $\{M' \in \mathsf{LDB}(\mathbf{V}) \mid M' \leq_{\mathbf{D}} M\}$ is a finite set.

It is difficult to envision any realistic database schema which would not satisfy the finite submodels condition. Even though there may be an infinite number of possible data items, a real database contains only finitely many of them. Clearly, any schema $\mathbf{D}$ which satisfies the finite submodels condition satisfies the descending chain condition for any $S \subseteq \mathsf{LDB}(\mathbf{D})$, and so in particular any of its views satisfies the local descending chain condition. Thus, for all practical purposes, there is no loss of generality in assuming that a view satisfies the local descending chain condition. The next lemma establishes that this condition is enough to ensure that transitive closure implies strong transitivity.

5.17 Lemma *If $\mathfrak{P}$ is transitively closed and Γ satisfies the local descending chain condition, then $\mathfrak{P}$ is strongly transitive.*

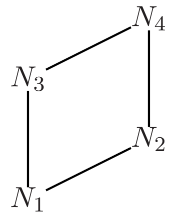
PROOF: Let $(N_1, N_2), (N_2, N_3), (N_1, N_3) \in \mathfrak{P}$, and let $M \in \gamma^{-1}(N_1)$. It is immediate that $\mathsf{RefSeq}\langle\Gamma, \mathfrak{P}\rangle(\langle N_1, N_3\rangle, M) \leq_{\mathbf{D}} \mathsf{RefSeq}\langle\Gamma, \mathfrak{P}\rangle(\langle N_1, N_2, N_3\rangle, M)$, since $\mathsf{RefSeq}\langle\Gamma, \mathfrak{P}\rangle(\langle N_1, N_3\rangle, M) = \inf(\{M' \in \gamma^{-1}(N_m) \mid M \leq_{\mathbf{D}} M'\}$. Now define the Γ-*height* of M to be the length of the longest descending chain $M_h \leq_{\mathbf{D}} M_{h-1} \leq_{\mathbf{D}} M_2 \leq_{\mathbf{D}} M$ of distinct elements in $\gamma^{-1}(N_1)$. It follows directly from 5.6(a) that the associations $M \mapsto \mathsf{RefSeq}\langle\Gamma, \mathfrak{P}\rangle(\langle N_1, N_2, N_3\rangle, M)$ and $M \mapsto \mathsf{RefSeq}\langle\Gamma, \mathfrak{P}\rangle(\langle N_1, N_3\rangle, M)$ are each isomorphisms from $\gamma^{-1}(N_1)$ to $\gamma^{-1}(N_3)$. In particular, if M has Γ-height h, then so too do $\mathsf{RefSeq}\langle\Gamma, \mathfrak{P}\rangle(\langle N_1, N_2, N_3\rangle, M)$ and $\mathsf{RefSeq}\langle\Gamma, \mathfrak{P}\rangle(\langle N_1, N_3\rangle, M)$. Since $\mathsf{RefSeq}\langle\Gamma, \mathfrak{P}\rangle(\langle N_1, N_3\rangle, M) \leq_{\mathbf{D}} \mathsf{RefSeq}\langle\Gamma, \mathfrak{P}\rangle(\langle N_1, N_2, N_3\rangle, M)$, the only way that this can occur is for $\mathsf{RefSeq}\langle\Gamma, \mathfrak{P}\rangle(\langle N_1, N_3\rangle, M) = \mathsf{RefSeq}\langle\Gamma, \mathfrak{P}\rangle(\langle N_1, N_2, N_3\rangle, M)$. $\square$

From a mathematical point of view, one could just as well use an ascending chain condition in the above result; chains in both directions are needed for the correspondence to fail. However, such an ascending chain condition is not particularly interesting from a database point of view, since many practical models do not possess it. (In many schemata, one can always add extra data to an existing instance.)

The result of 5.17 will be used in the final results 5.23 and 5.31. However, in the intermediate results given below, a more direct assumption of strong transitivity will be made, with the understanding that it may always be replaced with the local descending chain condition.

The following simple but very useful lemma applies in the context that $\mathfrak{P}$ is strongly transitive.

5.18 Lemma *Assume that* $\mathfrak{P}$ *is strongly transitive, and let* $N_1, N_2, N_3, N_4 \in \mathsf{LDB}(\mathbf{V})$ *have the property that* $(N_1, N_2), (N_1, N_3), (N_2, N_4), (N_3, N_4) \in \mathfrak{P}$. *For any* $i, j \in \{1, 2, 3, 4\}$ *with* $i \neq j$*, let* σ_1 *and* σ_2 *be the two cycle-free paths from* N_i *to* N_j *in the graph to the right. Then, for any* $M \in \gamma^{-1}(N_i)$, $\mathsf{RefSeq}\langle\Gamma, \mathfrak{P}\rangle(\sigma_1, M) = \mathsf{RefSeq}\langle\Gamma, \mathfrak{P}\rangle(\sigma_2, M)$.



Proof: First of all, if $\{i, j\} = \{1, 4\}$, then the proof is immediate from the assumption of strong transitivity. If $i = 3$, $j = 4$, then 5.6(b) and (c) may be used to show that $\{\langle N_3, N_1\rangle \cdot \langle N_1, N_3, N_4\rangle, \langle N_3, N_4\rangle\}$ and $\{\langle N_3, N_1\rangle \cdot \langle N_1, N_2, N_4\rangle, \langle N_3, N_1, N_2, N_4\rangle\}$ are each result equivalent pairs, whence $\langle N_3, N_4\rangle$ and $\langle N_3, N_1, N_2, N_4\rangle$ are result equivalent. The other cases in which one path has length one and the other length three are handled similarly. Finally, if $i = 3$ and $j = 2$, then 5.6(b) and (d) may be used to show that $\{\langle N_3, N_4\rangle \cdot \langle N_4, N_2\rangle, \langle N_3, N_4, N_2\rangle\}$ and $\{\langle N_3, N_1, N_2, N_4\rangle \cdot \langle N_4, N_2\rangle, \langle N_3, N_1, N_2\rangle\}$ are result equivalent pairs, whence $\langle N_3, N_4, N_2\rangle$ and $\langle N_3, N_1, N_2\rangle$ are view equivalent. Similarly, $\langle N_2, N_1, N_3\rangle$ and $\langle N_2, N_4, N_3\rangle$ are view equivalent. This covers all of the cases, whence the result. □

In a lifting of an update sequence $\sigma = \langle N_1, N_2, \ldots, N_m\rangle$, each element is "lifted" by joining it with a fixed element N in the same orbit, to yield a new update sequence $\langle \sup(\{N_1, N\}), \sup(\{N_2, N\}), \ldots, \sup(\{N_m, N\})\rangle$. As shall be seen, this is a very useful tool in establishing that a view admits unique updates.

5.19 Mutual lifting and the lifting of a view update sequence

(a) A subset $\{N_1, N_2\} \subseteq \mathsf{LDB}(\mathbf{V})$ is called a *mutual lifting pair* (relative to $\mathfrak{P}$) if $\sup(\{N_1, N_2\})$ exists in $\mathsf{LDB}(\mathbf{V})$ and both $(N_1, \sup(\{N_1, N_2\})), (N_2, \sup(\{N_1, N_2\})) \in \mathfrak{P}$.

(b) A pair of nonempty sets $\{S_1, S_2\} \subseteq \mathbf{2}^{\mathfrak{P}}$ is a *mutual lifting family* if each set of the form $\{N_1, N_2\}$ with $N_1 \in S_1$ and $N_2 \in S_2$ is a mutual lifting pair.

(c) Let $N \in \mathsf{LDB}(\mathbf{V})$. The view update sequence $\sigma = N_1, N_2, \ldots N_m$ *lifts along* N if $\{\{N\}, \{N_1, N_2, \ldots, N_m\}\}$ is a mutual lifting family. In that case, the sequence $\mathsf{Lift}\langle\Gamma, \mathfrak{P}\rangle(\sigma, N) = \langle \sup(\{N_1, N\}), \sup(\{N_2, N\}), \ldots, \sup(\{N_m, N\})\rangle$ is called the *lifting of* σ *along* N.

(d) $\mathfrak{P}$ *admits liftings* if for every view update sequence σ from $\mathfrak{P}$ there is an $N \in \mathsf{LDB}(\mathbf{V})$ with the property that σ lifts along N.

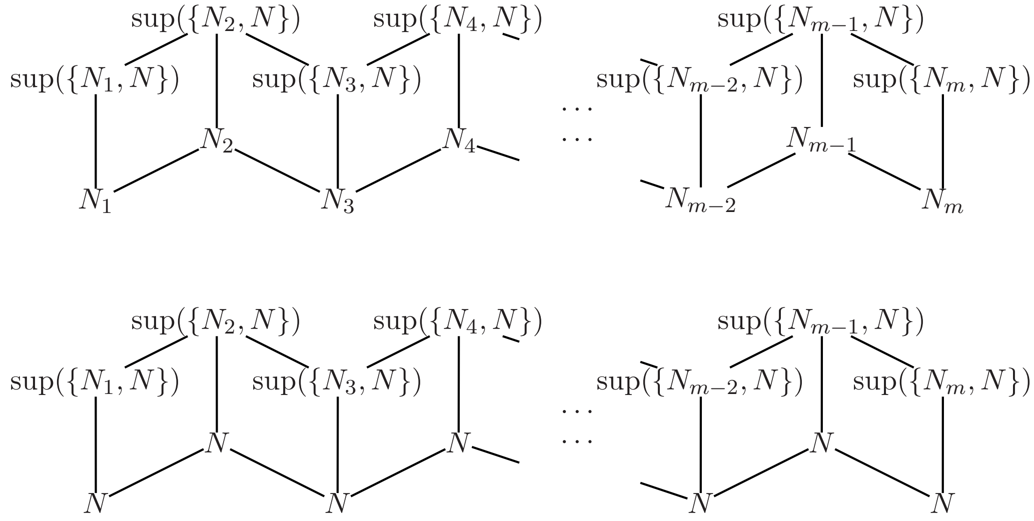


Figure 5: A graphical depiction of lifting

Figure 5 depicts graphically the idea of lifting. As shown in 5.21, these diagrams "commute" in the sense that the final value of a reflected update sequence depends only upon the first and last elements, and not upon the path. However, it is first necessary to establish that the lifted view update sequence is itself a view update sequence.

5.20 Lemma *Assume that $\mathfrak{P}$ is transitively complete.*

(a) *If $L = \{\{N\}, \{N_1, N_2\}\}$ is a mutual lifting family with $(N_1, N_2) \in \mathfrak{P}$, then $(\sup(\{N_1, N\}), \sup(\{N_2, N\})) \in \mathfrak{P}$ as well.*

(b) *Every lifting $\mathsf{Lift}\langle\Gamma, \mathfrak{P}\rangle(\sigma, N)$ of a view update sequence σ is itself a view update sequence.*

PROOF: (a) Since $(N_1, N_2), (N_2, \sup(\{N_2, N\})) \in \mathfrak{P}$, it follows from transitive closure that $(N_1, \sup(\{N_2, N\})) \in \mathfrak{P}$. Since $\sup(\{N_1, N\}) \leq_{\mathbf{v}} \sup(\{N_2, N\})$, it follows from transitive completeness that $(\sup(\{N_1, N\}), \sup(\{N_2, N\})) \in \mathfrak{P}$, as required. Part (b) now follows directly from (a). □

5.21 Lemma *Assume that $\mathfrak{P}$ is strongly transitive and transitively complete, and let $L = \{\{N\}, \{N_1, N_2, \ldots, N_m\}\}$ be a mutual lifting family relative to $\mathfrak{P}$. Let σ_1 and σ_2 be any two cycle-free paths of elements from either of the diagrams of Fig. 5 with the same start and end points. Then σ_1 and σ_2 are update equivalent.*

PROOF: The proof follows directly from 5.18, and 5.20. □

5.22 Proposition *If $\mathfrak{P}$ is strongly transitive, transitively complete, and Γ admits liftings, then $\mathfrak{P}$ defines unique updates.*

PROOF: Let $\sigma_1 = \langle N_{11}, N_{12}, \ldots, N_{1m_1}\rangle$ and $\sigma_2 = \langle N_{21}, N_{22}, \ldots, N_{2m_2}\rangle$ be view update sequences with the property that $N_{11} = N_{21}$ and $N_{1m_1} = N_{2m_2}$. Define $\sigma = \sigma_1 \cdot \sigma_2^R$.

For compatibility with the diagrams of Fig. 5, write $\sigma = \langle N_1, N_2, \ldots, N_m\rangle$, noting that $N_1 = N_m = N_{11} = N_{21}$. Now let N be such that $\{\{N\}, \{N_1, N_2, \ldots, N_m\}\}$ is a mutual lifting family. The proof is achieved by pasting together the two diagrams of Fig. 5 and invoking 5.21. First of all, from the lower diagram of Fig. 5, it may be concluded that $\sigma' = \langle \sup(\{N_1, N\}), \sup(\{N_2, N\}), \ldots, \sup(\{N_m, N\})\rangle$ represents the identity update, in the sense that for any $M \in \gamma^{-1}(\sup(\{N_1, N\}))$, $\mathsf{RefSeq}\langle\Gamma, \mathfrak{P}\rangle(\sigma', M) = M$. From the upper diagram it then follows that this update may be "lowered" from σ' to σ. Thus, $\sigma = \sigma_1 \cdot \sigma_2^R$ must be equivalent to $\langle N_{11}, N_{11}\rangle$, the identity update. Since $\sigma_1 \cdot \sigma_1^R$ is certainly equivalent to $\langle N_{11}, N_{11}\rangle$, it follows from 5.6(d) that σ_2^R and σ_1^R are result equivalent, whence σ_1 and σ_2 are as well. □

It is finally possible to obtain the alternate to 5.10, this time based upon the existence of liftings rather than directly specifying that it define unique updates.

5.23 Theorem — alternate partial characterization of update strategies *Define $T = \mathsf{OrdUpd}(\mathfrak{P})$. If $\mathfrak{P}$ is transitively closed, transitively complete, admits liftings, and satisfies the local descending chain condition, then $\rho = \mathsf{RawUpdStr}\langle\Gamma, \mathfrak{P}\rangle$ satisfies conditions (upt:1) through (upt:7) of 3.1.*

PROOF: This follows immediately from 5.10, 5.17, and 5.22. □

Before turning to the question of an alternate characterization of the strong orbital linking condition, it is appropriate to provide an additional consequence of the use of lifting. Basically, it says that any update sequence composed of insertions and deletions may be replaced by an equivalent one which involves no more than four basic insertion and deletion operations.

5.24 Proposition — four basic updates suffice *If $\mathfrak{P}$ admits unique updates and liftings, then every order-based update is realizable as a sequence of at most two insertions and two deletions.*

PROOF: Let $\sigma = N_1, N_2, \ldots N_m$ be the update sequence to be realized, and let $N \in \mathsf{LDB}(\mathbf{V})$ have the property that σ lifts along N. The sequence of updates shown in the diagram to the right has the desired property. □

$$\begin{tikzcd}[column sep=small] & \sup(\{N_1, N\}) \arrow[dl, dash] \arrow[dr, dash] & & \sup(\{N_m, N\}) \arrow[dl, dash] \arrow[dr, dash] & \\ N_1 & & N & & N_m \end{tikzcd}$$

The strong orbital linking is often difficult to verify directly. Therefore, it is useful to seek conditions which are more easily observed in examples, yet also guarantee that the update reflection property is satisfied. In that which follows, one such approach is developed.

5.25 Generalized deletions

(a) The view Γ is *downward projective* if every pair $(N_1, N_2) \in \mathsf{LDB}(\mathbf{V}) \times \mathsf{LDB}(\mathbf{V})$ with $N_1 \leq_{\mathbf{V}} N_2$ is downward projective in the sense of 5.1(b). Note that this condition must be satisfied by every such pair (N_1, N_2), and not just those which lie in $\mathfrak{P}$.

(b) The view Γ is *strongly downward transitive* if it is downward projective and, for every triple $N_1, N_2, N_3 \in \mathsf{LDB}(\mathbf{V})$ with $N_1 \leq_{\mathbf{V}} N_2 \leq_{\mathbf{V}} N_3$, and every $M \in \gamma^{-1}(N_3)$, $\mathsf{Delete}_{\mathbf{V}}(M, N_1) = \mathsf{Delete}_{\mathbf{V}}(\mathsf{Delete}_{\mathbf{V}}(M, N_2), N_1)$. This is a generalization, in the downward (deletion) direction, of strong transitivity (5.16(a)).

(c) The view Γ *supports generalized deletions* relative to $\mathfrak{P}$ if it supports unique updates and, for any $(N_{11}, N_{12}), (N_{21}, N_{22}) \in \mathsf{OrdUpd}(\mathfrak{P})$ with $N_{11} \leq_{\mathbf{V}} N_{21}$ and $N_{12} \leq_{\mathbf{V}} N_{22}$, and any $M \in \gamma^{-1}(N_{21})$, $\mathsf{Delete}_{\mathbf{V}}(\mathsf{RawUpdStr}\langle\Gamma, \mathfrak{P}\rangle(M, N_{22}), N_{12}) = \mathsf{RawUpdStr}\langle\Gamma, \mathfrak{P}\rangle(\mathsf{Delete}_{\mathbf{V}}(M, N_{11}), N_{12})$. This may be visualized with the commutative diagram shown below.

$$\begin{tikzcd}
\mathsf{LDB}(\mathbf{D}) \arrow[r, "\mathsf{RawUpdStr}\langle\Gamma,\mathfrak{P}\rangle(-,N_{22})"] \arrow[d, "\mathsf{Delete}_{\mathbf{V}}(-,N_{11})"] & \mathsf{LDB}(\mathbf{D}) \arrow[d, "\mathsf{Delete}_{\mathbf{V}}(-,N_{12})"] \\
\mathsf{LDB}(\mathbf{D}) \arrow[r, "\mathsf{RawUpdStr}\langle\Gamma,\mathfrak{P}\rangle(-,N_{12})"] & \mathsf{LDB}(\mathbf{D})
\end{tikzcd}$$

The idea behind supporting generalized deletions is that for $N_1 \leq_{\mathbf{V}} N_2$ and $M_1 \in \gamma(N_1)$, deletion from M_1 to a state in $\gamma^{-1}(N_2)$ is well defined and well behaved, even if this update is not reversible. The idea of generalized deletions is closely related to the notion of updates with decreasing complements, in the sense of [9], although here it is used as a tool to characterize views which admit update strategies in the sense of 3.1, rather than as a liberal form of update to be allowed on its own merits.

5.26 Examples — downward projection and generalized deletion Shown in Fig. 6 are some examples related to downward projection. The views depicted in both

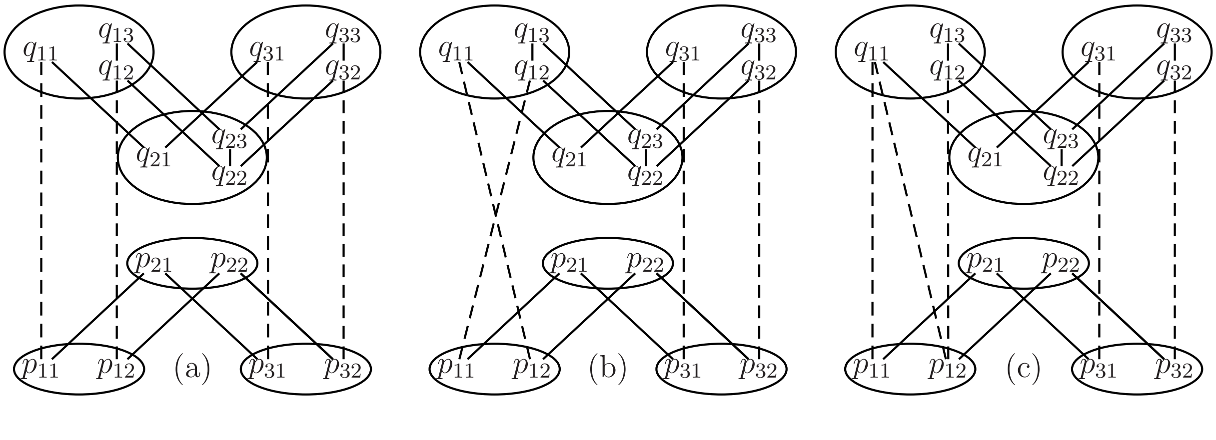


Figure 6: Examples related to downward projection and generalized deletion

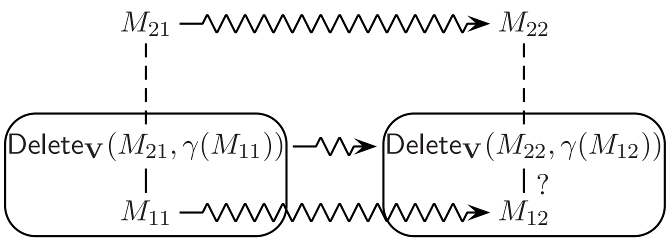
diagrams (a) and (b) are downward projective, while (c) is not. In (c), there is no $M \in \{p_{11}, p_{12}\}$ with the property that it is the greatest element in that set which is smaller than q_{11}.

It is nonetheless clear that (a) depicts a more desirable situation than (b). Indeed, (a) is both strongly downward transitive and supports generalized deletions, while (b) has neither property. In terms of a generalized deletion, diagram (b) suffers from the same problem that diagram (e) of Fig. 3 does for an ordinary deletion; namely, that different results are obtained, depending upon the path taken. Although the updates $q_{11} \to p_{12} \to p_{22} \to p_{32}$ and $q_{11} \to q_{21} \to q_{31} \to p_{31}$ are not allowed formally, to establish necessary conditions for the update reflection property to hold, the schema must not contain such pairs of update paths which lead to different elements in the same equivalence class.

In any case, support for generalized deletions, in the presence of unique updates, is sufficient to ensure that the strong orbital linking condition is satisfied, as shown below.

5.27 Proposition *If Γ defines unique updates and supports generalized deletions relative to $\mathfrak{P}$, then it satisfies the strong orbital linking condition.*

PROOF: Let $M_{11}, M_{21} \in \mathsf{LDB}(\mathbf{D})$ with $M_{11} \leq M_{21}$, and let $M_{12} \in \mathsf{Orbit}\langle\Gamma, \mathfrak{P}\rangle(M_{11})$, $M_{22} \in \mathsf{Orbit}\langle\Gamma, \mathfrak{P}\rangle(M_{21})$ with $\gamma(M_{12}) \leq_{\mathbf{V}} \gamma(M_{22})$. This situation may be visualized with the diagram to the right. The zigzag horizontal lines represent true updates, while the straight vertical lines represent ordering in $\mathsf{LDB}(\mathbf{D})$. The upper two vertical lines also represent generalized deletions. The rounded rectangles contain states of $\mathsf{LDB}(\mathbf{D})$ which lie in the same equivalence class of Γ. The diagram defined by the upper four nodes is just an instantiation of the diagram of 5.25, and so commutes. In the lower diagram, it is the case that the left vertical line is a correct representation of the ordering, since $M_{11} \leq_{\mathbf{D}} M_{21}$ is given, and $M_{11} \leq_{\mathbf{D}} \mathsf{Delete}_{\mathbf{V}}(M_{21}, \gamma(M_{11}))$ by definition of $\mathsf{Delete}_{\mathbf{V}}$. The key question is whether the vertical line in the right rectangle labelled with a question mark is actually a correct representation of the ordering; *i.e.*, whether $M_{12} \leq \mathsf{Delete}_{\mathbf{V}}(M_{22}, \gamma(M_{12}))$. However, this follows easily from 5.13(b), since M_{11} and $\mathsf{Delete}_{\mathbf{V}}(M_{21}, \gamma(M_{11}))$ lie in parallel orbits. Thus, $M_{12} \leq_{\mathbf{D}} M_{22}$, and so the strong orbital linking condition is satisfied. □



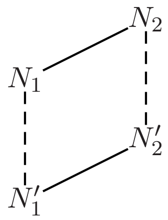
It is not always easy to verify support for generalized deletions directly. Fortunately, there is a useful characterization in terms of lifting, which is developed next.

5.28 Projection of liftings The view Γ *projects liftings* if for every pair $\{\{N\}, \{N_1, N_2\}\}$ and $\{\{N'\}, \{N'_1, N'_2\}\}$ of mutual lifting families with $N'_1 \leq_{\mathbf{V}} N_1$ and $N'_2 \leq_{\mathbf{V}} N_2$, there is an $N'' \in \mathsf{LDB}(\mathbf{V})$ with the property that $N'' \leq_{\mathbf{V}} N$ and $\{\{N''\}, \{N'_1, N'_2\}\}$ is also a mutual lifting family.

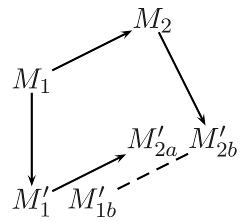
The following lemma is a limited extension of 5.18 to generalized deletions, and applies to updates which do not follow a path up the vertical lines of the trapezoid.

5.29 Lemma *Assume that Γ is strongly downward transitive, and let $N_1, N_2, N_3, N_4 \in$*

$\mathsf{LDB}(\mathbf{V})$ *with* $N_1 \leq_{\mathbf{V}} N_2 \leq_{\mathbf{V}} N_4$ *and* $N_1 \leq_{\mathbf{V}} N_3 \leq_{\mathbf{V}} N_4$, *as shown in the diagram to the right. Assume further that both* (N_1, N_2), $(N_1', N_2') \in \mathfrak{P}$, *with no such assumption made for* (N_1', N_1) *or* (N_2', N_2). *Then the two paths from* N_1 *to* N_2', *as well as the two paths from* N_2 *to* N_1', *are update equivalent in the precise sense that for any* $M_1 \in \gamma^{-1}(N_1)$, $\mathsf{Delete}_{\mathbf{V}}(\mathsf{Insert}_{\mathbf{V}}(M_1, N_2), N_2') = \mathsf{Insert}_{\mathbf{V}}(\mathsf{Delete}_{\mathbf{V}}(M_1, N_1'), N_2')$, *and for any* $M_2 \in \gamma^{-1}(N_2)$, $\mathsf{Delete}_{\mathbf{V}}(\mathsf{Delete}_{\mathbf{V}}(M_2, N_1), N_1') = \mathsf{Delete}_{\mathbf{V}}(\mathsf{Delete}_{\mathbf{V}}(M_2, N_2'), N_1')$.

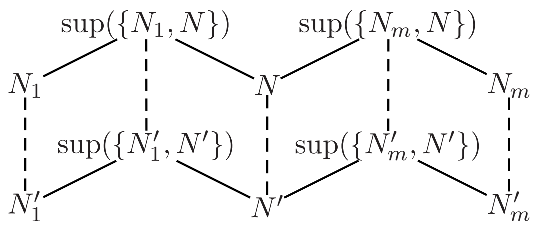


PROOF: The equivalence of the two paths from N_2 to N_1' follows immediately from the definition of strong downward transitivity 5.25(b). For the two paths from N_1 to N_2', consider the diagram shown to the right. $M_1 \in \mathbf{D}$ is the start state of N_1, and the update to N_2 and then N_2' is along M_2 and then M_{2b}', while the update along N_1' and then N_2' is along M_1' and then M_{2a}'. Add to the diagram the update M_{2b}' to N_1', yielding M_{1b}'. Invoking the property of strong downward transitivity, it follows that both M_1' and M_{1b}' are $\mathsf{Delete}_{\mathbf{V}}(M_2, N_1')$, whence they are equal. Since $(N_1', N_2') \in \mathfrak{P}$, it follows from 5.6(a) that $M_{2a}' = M_{2b}'$ as well, completing the proof. □



5.30 Proposition *Let* $\mathfrak{P}$ *be transitively closed and transitively complete, and suppose that* Γ *admits and projects liftings and is strongly downward transitive. Then* Γ *supports generalized deletions.*

PROOF: Let $\sigma = N_1, N_2, \ldots N_m$ and $\sigma' = N_1, N_2, \ldots N_m'$ be update sequences, and suppose that $N_1' \leq_{\mathbf{V}} N_1$ and $N_2' \leq_{\mathbf{V}} N_2$. In view of 5.24, there are $N, N' \in \mathsf{LDB}(\mathbf{V})$, with the property that $\{\{N\}, \{N_1, N_2\}\}$ and $\{\{N'\}, \{N_1', N_2'\}\}$ are mutual lifting families, and so the parts of the diagram to the right connected by solid lines commute. Under the assumption of projected lifting, N and N' may be chosen so that $N' \leq_{\mathbf{V}} N$. It then follows that first, third and fifth dashed vertical lines are correct indications of order. The second and fourth (between the $\sup(-)'s$) are clearly valid from general principles of suprema in posets. Therefore, all lines indicate valid order relationships, with the solid ones indicating connections between strong order pairs as well. Now, application of 5.29 to each trapezoid yields that the two paths from N_1 to N_m' yield identical values, so that Γ supports generalized deletions. □



It is finally possible to provide a unified alternative to 5.15.

5.31 Theorem — alternate characterization of update strategies *Let* $\mathfrak{P}$ *be transitively closed, transitively complete, and suppose that* Γ *admits and projects liftings, is strongly downward transitive, and satisfies the local descending chain condition. Then* $\rho = \mathsf{RawUpdStr}\langle\Gamma, \mathfrak{P}\rangle$ *is an update strategy in the sense of 3.1.*

PROOF: 5.22 above guarantees that $\mathfrak{P}$ defines unique updates, while 5.27 establishes that it satisfies the strong orbital linking condition. The result then follows from the general characterization theorem 5.15. □

6. Direct Construction of Update Strategies in the Relational Framework

In this section, the tools which were developed in the previous section are brought to bear upon views in a most classical framework, projections of a single relation constrained by equality generating dependencies (EGD's). It is shown that for views which retain all but one column, if there is a universal set of updates, then it must be those defined by the classical case in which the complement is also a projection, with the common columns held constant. For views which discard more than one column, the situation is more complex, but valuable insights are provided for that context as well.

6.1 Notational conventions Throughout this section, $\mathbf{U}$ will be taken to be a finite set of attributes, and $\mathbf{W} \subseteq \mathbf{U}$. To each $A \in \mathbf{U}$, there is a countably infinite set $\mathsf{Dom}(A)$ of constants for that domain.

Although the relational framework is well known, it is nonetheless important to establish some notation and representational techniques which are specific to the framework used in this paper. The following summary is present in that capacity. For further information on concepts within logic, [19] is a comprehensive reference.

6.2 The logic of a single-relation schema The *logic* $\mathcal{L}_{\mathbf{U}}$ of $\mathbf{U}$ has a sort Σ_A for each $A \in \mathbf{U}$, and a single $\mathbf{U}$-ary relation symbol, denoted by $R_{\mathbf{U}}$. For each $A \in \mathbf{U}$, it also has an equality relation $=_A$. When no confusion can result, equality will be denoted simply by $=$, although it is to be understood that only terms of the same sort may be compared. The constant symbols of sort Σ_A are precisely the members of the countably infinite set $\mathsf{Dom}(A)$. In addition, for each $A \in \mathbf{U}$, there is a countable set $\mathsf{Vars}(A)$ of variables of sort Σ_A. There are no non-nullary function symbols; thus, the terms $\mathsf{Terms}(A)$ of this logic which are of sort Σ_A are precisely the members of $\mathsf{Dom}(A) \cup \mathsf{Vars}(A)$. The notation $\mathsf{Dom}(\mathbf{U})$ (resp. $\mathsf{Vars}(\mathbf{U})$, resp. $\mathsf{Terms}(\mathbf{U})$) denotes $\bigcup\{\mathsf{Dom}(A) \mid A \in \mathbf{U}\}$ (resp. $\bigcup\{\mathsf{Vars}(A) \mid A \in \mathbf{U}\}$, resp. $\bigcup\{\mathsf{Terms}(A) \mid A \in \mathbf{U}\}$).

A $\mathbf{U}$-*atom* (also called a $\mathbf{U}$-*tuple*) is a function $t : \mathbf{U} \to \mathsf{Terms}(\mathbf{U})$ with the property that $t[A] \in \mathsf{Terms}(A)$ for each $A \in \mathbf{U}$. $t[\mathbf{W}]$ denotes the $\mathbf{W}$-term which is obtained by retaining the values associated with the attributes in $\mathbf{W}$ and discarding the others. $\mathsf{Tuples}(\mathbf{U})$ denotes the set of all $\mathbf{U}$-tuples. It will be assumed that there is an implicit ordering on the attributes in $\mathbf{U}$, so that positional indexing may also be used. This ordering is usually specified by listing the attributes in sequence. For example, in the context of $R_{\mathbf{U}}[ABC]$, the ordering is (A, B, C), and a corresponding atom might be represented as $R_{\mathbf{U}}(a_0, b_0, c_0)$. Note, however, that the tuple notation is also available, so if $t = R_{\mathbf{U}}(a_0, b_0, c_0)$, then $t[BC] = (b_0, c_0)$, *etc.* $\mathbf{U}$-atoms never include atoms which are based upon an equality relation, so that, for example $(v_o^A =_A v_1^A)$ is always called an *equality atom*; it is never referred to as a $\mathbf{U}$-atom. When the context is clear, the terminology $\mathbf{U}$-atom (resp. $\mathbf{U}$-tuple) will be shortened to just *atom* (resp. *tuple*).

Well-formed formulas are built up from atoms in the usual way; it is always assumed that formulas are *well typed* in the sense that terms only occur in the appropriate positions in atoms, and equality atoms only occur between two terms of the same sort. Sentences are well-formed formulas with no free variables. Structures and models may also be constructed in the usual way; for each sort Σ_A there is a set $\mathsf{Univ}(A)$ of values; a structure is then a relation whose A-th column contains values from $\mathsf{Univ}(A)$. Models are defined in the usual way; if Ψ is a set of sentences in the language of $\mathcal{L}_{\mathbf{U}}$, the set of all models of Ψ is denoted $\mathsf{Mod}(\Psi)$. The symbol $\models$ denotes semantic entailment; if Ψ is a set of sentences and ψ is a sentence, then $\Psi \models \psi$ holds iff $\Psi \cup \{\neg\psi\}$ is not satisfiable. For Ψ_1 and Ψ_2 sets of sentences, $\Psi_1 \models \Psi_2$ means that $\Psi_1 \models \psi$ for each $\psi \in \Psi_2$. It is important to note that $\models$ represents a purely semantic notion; it is not associated with any particular deductive method.

In the context of this work, it is more convenient to define databases as sets of atoms than as models of a set of constraints. Specifically, if $t[A] \in \mathsf{Dom}(A)$ for each $A \in \mathbf{U}$, then t is called a *ground atom* on $\mathbf{U}$. The set of all ground atoms (resp. atoms) on $\mathbf{U}$ is denoted $\mathsf{GndAtoms}(\mathbf{U})$ (resp. $\mathsf{Atoms}(\mathbf{U})$). The set of *databases* on $\mathbf{U}$ is denoted $\mathsf{DB}(\mathbf{U})$, and consists of all finite sets of ground atoms on $\mathbf{U}$. For example, if $\mathbf{U} = \{A, B, C\}$, the relation $\{(a_0, b_0, c_0), (a_1, b_1, c_0), (a_1, b_2, c_2)\}$ will be represented logically by the following set of ground atoms. $\{R_{\mathbf{U}}(a_0, b_0, c_0), R_{\mathbf{U}}(a_1, b_1, c_0), R_{\mathbf{U}}(a_1, b_2, c_2)\}$. As an abuse of notation, the same symbol (*e.g.*, M) shall often be used to represent both sets. These databases are ordered via subset inclusion of the underlying ground atoms, in the obvious way.

It will be necessary on occasion to be able to work with the product of two databases. Because databases are regarded as sets of ground atoms rather than as relations, some care is necessary. Specifically, define $\mathsf{Dom}_\Delta(\mathbf{U}) = \bigcup\{\mathsf{Dom}(A) \times \mathsf{Dom}(A) \mid A \in \mathbf{U}\}$, and let $\xi : \mathsf{Dom}_\Delta(\mathbf{U}) \to \mathsf{Dom}(\mathbf{U})$ be a bijection with the property that $\xi(\mathsf{Dom}(A) \times \mathsf{Dom}(A)) = \mathsf{Dom}(A)$. In other words, ξ provides, for each $A \in \mathbf{U}$, a bijective correspondence between $\mathsf{Dom}(A) \times \mathsf{Dom}(A)$ and $\mathsf{Dom}(A)$. For $t_1, t_2 \in \mathsf{GndAtoms}(\mathbf{U})$, the product $t_1 \otimes t_2 \in \mathsf{GndAtoms}(\mathbf{U})$ relative to ξ has $(t_1 \otimes t_2)[A] = \xi(t_1[A], t_2[A])$. For $M_1, M_2 \in \mathsf{DB}(\mathbf{U})$, the *product* (relative to ξ) $M_1 \otimes M_2 = \{t_1 \otimes t_2 \mid t_1 \in M_1 \text{ and } t_2 \in M_2\}$. Since the particular choice of ξ will never be of consequence, it is assumed to be present implicitly, without specific mention.

Let Ψ be a set of constraints; *i.e.*, a set of sentences in the language $\mathcal{L}_{\mathbf{U}}$, and let $M \in \mathsf{DB}(\mathbf{U})$. Define $\mathsf{Wff}(M)$ to be the conjunction of all atoms in M. The state M *satisfies* Ψ iff $\Psi \cup \{\mathsf{Wff}(M)\}$ is satisfiable. The set of all databases which satisfy Ψ is denoted $\mathsf{LDB}_\Psi(\mathbf{U})$.

An *extended database* for $\mathbf{U}$ is represented by a finite set of atoms, which need not all be ground atoms. The set of all extended databases for $\mathbf{U}$ is denoted $\mathsf{ExtDB}(\mathbf{D})$. The set of variables in $\mathsf{Vars}(\mathbf{U})$ which occur in $M \in \mathsf{ExtDB}(\mathbf{D})$ is denoted $\mathsf{DBVars}(M)$. If $M = \{t_1, t_2, \ldots, t_n\} \in \mathsf{ExtDB}(\mathbf{U})$, then the associated formula $\mathsf{Wff}(M) = (Q)(t_1 \wedge t_2 \wedge \ldots \wedge t_n)$, with Q a quantifier prefix consisting of the concatenation of declarations of the form $(\exists x)$, one for each variable $x \in \mathsf{DBVars}(M)$. For example, if $M = \{R_{\mathbf{U}}(a_0, b_0, v_3^C), R_{\mathbf{U}}(a_1, v_1^B, v_2^C), R_{\mathbf{U}}(a_1, v_1^B, c_2)\}$, then $\mathsf{DBVars}(M) = \{v_3^C, v_1^B, v_2^C\}$, and the associated formula is $(\exists v_3^C)(\exists v_1^B)(\exists v_2^C)(R_{\mathbf{U}}(a_0, b_0, v_3^C) \wedge R_{\mathbf{U}}(a_1, v_1^B, v_2^C) \wedge R_{\mathbf{U}}(a_1, v_1^B, c_2))$. In harmony with the case of ordinary databases, an extended database M *satisfies* the set of constraints Ψ if $\Psi \cup \{\mathsf{Wff}(M)\}$ is satisfiable, and $\mathsf{ExtLDB}_\Psi(\mathbf{U})$ denotes the set of all members of $\mathsf{ExtDB}(\mathbf{M})$ which satisfy Ψ. Note that there can be inessential variations in the representation of an extended database. For example, $M_1 = \{R_{\mathbf{U}}(a_0, b_0, v_0^C)\}$ and $M_2 = \{R_{\mathbf{U}}(a_0, b_0, v_0^C), R_{\mathbf{U}}(a_0, b_0, v_1^C)\}$ represent the same database semantically, in the sense that $\mathsf{Wff}(M_1)$ and $\mathsf{Wff}(M_2)$ are equivalent. The proper way to impose an ordering on $\mathsf{ExtDB}(\mathbf{U})$ is as follows. For $M_1, M_2 \in \mathsf{ExtDB}(\mathbf{U})$, write $M_1 \preceq_{\mathbf{U}} M_2$ just in case $\mathsf{Wff}(M_1) \models \mathsf{Wff}(M_2)$. This is not a partial order, but rather only a preorder; $M_1 \preceq_{\mathbf{U}} M_2 \preceq_{\mathbf{U}} M_1$ iff $\mathsf{Wff}(M_1)$ and $\mathsf{Wff}(M_2)$ are logically equivalent.

Substitutions are defined as usual, and are required to be well typed. Specifically, a *substitution* is a function $\alpha : \mathsf{Vars}(\mathbf{U}) \to \mathsf{Terms}(\mathbf{U})$ with the property that $\alpha(\mathsf{Vars}(A)) \subseteq \mathsf{Terms}(A)$ for each $A \in \mathbf{U}$. The substitution α is *grounding* if $\alpha(v) \in \mathsf{Dom}(\mathbf{U})$ for each $v \in \mathsf{Vars}(\mathbf{U})$. $\alpha(t)$ denotes the application of the substitution α to the atom t; each occurrence of each variable $v \in \mathsf{Vars}(\mathbf{U})$ which occurs in t is replaced with the term $\alpha(v)$. For $M \in \mathsf{ExtDB}(\mathbf{U})$, $\alpha(M) = \{\alpha(t) \mid t \in M\}$.

Finally, a set of generic functions, including $\mathsf{Sort}(-)$, $\mathsf{Vars}(-)$, $\mathsf{Terms}(-)$, and $\mathsf{Constants}(-)$, may be applied to any object for which they make sense. The last three each return a set of values; for example, for $M \in \mathsf{DB}(\mathbf{U})$, $\mathsf{Constants}(M)$ returns the set of all constants symbols which occur in atoms of M. $\mathsf{Sort}(-)$ returns a single value, and may only be used in contexts in which that makes sense; $\mathsf{Sort}((a = b))$ returns the sort of the constants a and b which are compared in the equality.

6.3 Schemata and views in the universal context Let Φ be a set of sentences in the language of $\mathcal{L}_{\mathbf{U}}$. Formally, the *schema of* $\mathbf{U}$ *with constraints* Φ is denoted $\mathsf{Sch}(\mathbf{U}, \Phi)$, and has $\mathsf{LDB}_{\Phi}(\mathbf{U})$ as its legal states. Note that under this convention, the translation to the notation used earlier in this paper is $\mathsf{LDB}_{\Phi}(\mathbf{U}) = \mathsf{LDB}(\mathsf{Sch}(\mathbf{U}, \Phi))$. For simplicity, the relations $\leq_{\mathsf{Sch}(\mathbf{U},\Phi)}$ and $\preceq_{\mathsf{Sch}(\mathbf{U},\Phi)}$ will be denoted by $\leq_{\mathbf{U}}$ and $\preceq_{\mathbf{U}}$, respectively, since they do not depend upon the choice of Φ.

The database mapping $\pi_{\mathbf{W}} : \mathsf{ExtDB}(\mathbf{U}) \to \mathsf{ExtDB}(\mathbf{W})$ is defined in the obvious way; $\pi_{\mathbf{W}}(M) = \{t[\mathbf{W}] \mid t \in M\}$. The same symbol will be used when this function is restricted to the domain $\mathsf{LDB}_{\Phi}(\mathbf{U})$. It is essential to be able to create the view of $\mathsf{Sch}(\mathbf{U}, \Phi)$ whose states are precisely the members of $\pi_{\mathbf{W}}(\mathsf{LDB}_{\Phi}(\mathbf{U}))$. However, since a view mapping must be surjective, the ability to do this depends critically upon the possibility of finding a set Ψ of constraints on $\mathbf{W}$ with the property that $\pi_{\mathbf{W}}(\mathsf{LDB}_{\Phi}(\mathbf{U})) = \mathsf{LDB}_{\Psi}(\mathbf{W})$. In general this is a highly nontrivial question; however, within the context of the dependencies used in this section, it has a positive answer. The details are presented below.

6.4 Equality-generating dependencies An *equality generating dependency* (*EGD*) over $\mathbf{U}$ is a sentence in $\mathcal{L}_{\mathbf{U}}$ of the form

$$(\forall.))((t_1 \wedge t_2 \wedge \ldots \wedge t_n) \Rightarrow \varepsilon) \qquad \text{(egd1)}$$

in which the following conditions are satisfied:

(i) The entries in each t_i consist entirely of variables; there are no terms from $\mathsf{Dom}(\mathbf{U})$.

(ii) ε is an equality atom of the form $(v_i^A =_A v_j^A)$ for some $A \in \mathbf{U}$, with the restriction that each of these variables occurs in at least one of the t_i's.

The prefix $(\forall.)$ means that all variables are quantified universally. The atoms to the left of the $\Rightarrow$ sign are called the *antecedents*, and the equality atom ε to the right is called the *consequent*. For a specific rule φ, the set of all antecedents (resp. the consequent) of φ is denoted $\mathsf{Antecedents}(\varphi)$ (resp. $\mathsf{Consequent}(\varphi)$).

Sometimes it is convenient to use an alternate form of representation for an EGD, in which the equality constraints between the antecedents are moved to explicit equality atoms. Specifically, an EGD in *explicit equality form* is a sentence of the form

$$(\forall.))((t_1 \wedge t_2 \wedge \ldots \wedge t_n \wedge \varepsilon_1 \wedge \varepsilon_2 \wedge \ldots \varepsilon_m) \Rightarrow \varepsilon) \qquad \text{(egd2)}$$

in which the following conditions are satisfied:

(i) The entries in each t_i consist entirely of variables, and no variable occurs more than once throughout the t_i's.

(ii) Each of the ε_i's, as well as ε, is an equality atom of the form $(v_i^A =_A v_j^A)$ for some $A \in \mathbf{U}$, with the restriction that each of these variables occurs in at least one of the t_i's.

In this section, the only constraints which are considered are EGD's. Therefore, the notation Φ^+ will denote the set of all EGD's φ for which $\Phi \models \varphi$, and not the more general set of all constraints which follow from Φ.

The most important flavor of EGD is the ubiquitous functional dependency, or *FD* for short. For $\mathbf{B} \subseteq \mathbf{U}$ and $A \in \mathbf{U}$, the FD $\mathbf{B} \rightarrow A$ is represented by the egd $(t_1 \wedge t_2) \Rightarrow \varepsilon)$ with $t_1[\mathbf{B}] = t_2[\mathbf{B}]$, and with all other variables in t_1 and t_2 distinct. ε has the form $(t_1[A] = t_2[A])$. For example, on the schema $R[ABCD]$, the functional dependency $AB \rightarrow D$ is represented by the following formula.

$$(\forall.)((R_{\mathbf{U}}(v_0^A, v_0^B, v_0^C, v_0^D) \wedge R_{\mathbf{U}}(v_0^A, v_0^B, v_1^C, v_1^D)) \Rightarrow (v_0^D =_D v_1^D))$$

This same dependency in explicit equality form is

$$(\forall.)((R_{\mathbf{U}}(v_0^A, v_0^B, v_0^C, v_0^D) \wedge R_{\mathbf{U}}(v_1^A, v_1^B, v_1^C, v_1^D) \wedge (v_0^A =_A v_1^A) \wedge (v_0^B =_B v_1^B)) \Rightarrow (v_0^D =_D v_1^D))$$

The FD $\mathbf{B} \rightarrow A$ is *nontrivial* if $A \notin \mathbf{B}$.

The following simple but key result is immediate from the definition of FD.

6.5 Observation *An EGD φ is equivalent to a functional dependency iff it is equivalent to an EGD with exactly two antecedent tuples.* $\square$

The following results are all well known, but are so fundamental to the results developed here that they are recorded explicitly, for reference.

6.6 Proposition — basic closure properties of EGD's *Let Φ be a set of EGD's on $\mathbf{U}$.*

(a) *There is a set Ψ of EGD's on $\mathbf{W}$ with the property that $\mathsf{LDB}_\Psi(\mathbf{W}) = \pi_{\mathbf{W}}(\mathsf{LDB}_\Phi(\mathbf{U}))$. This set need not be finite, even if Φ is.*

(b) *If* $M \in \mathsf{LDB}_\Phi(\mathbf{U})$ *and* $M' \subseteq M$ *is any subset of* M *whatever, then* $M' \in \mathsf{LDB}_\Phi(\mathbf{U})$ *as well.*

(c) *If* $M_1, M_2 \in \mathsf{LDB}_\Phi(\mathbf{U})$*, then* $M_1 \otimes M_2 \in \mathsf{LDB}_\Phi(\mathbf{U})$ *as well.*

(d) *If* $M_1, M_2 \in \mathsf{DB}(\mathbf{\Phi})$ *and* $M_1 \otimes M_2 \in \mathsf{LDB}_\Phi(\mathbf{U})$*, then* $M_1, M_2 \in \mathsf{LDB}(\mathbf{U})$*.*

PROOF: For (a), consult [8, 6.1]. For an example of a schema constrained by functional dependencies whose projection is an infinite set of EGD's, see [17, 4.1] or [13]. For (b), note that EGD's can only constrain existing tuples; they have no capacity to generate new ones. Therefore, removing tuples cannot lead to a structure which does not satisfy an EGD satisfied by the original one. Points (c) and (d) are consequences of the general fact that quasivarieties (*i.e.*, classes of models which are constrained by universal Horn sentences) are closed under products and substructures [19, 25.13]. These facts have also been noted in the database context by Fagin [8, 2.1], where a set of dependencies whose models are closed under (general, not necessarily finite) products is termed *upward faithful*, while closure under projection is termed *downward faithful*. In this paper, attention is restricted to finite (in fact, binary) products because that is all that is needed, and because only finite models are considered. □

6.7 Notational conventions Throughout the remainder of this section, Φ will denote an arbitrary set of EGD's over $\mathbf{U}$. The following notational simplifications are furthermore followed.

(a) The set Ψ of constraints over $\mathbf{W}$ guaranteed by 6.6(a) is denoted $\pi_{\mathbf{W}}(\Phi)$, and is called the $\mathbf{W}$-*projection* of Φ.

(b) The view of $\mathsf{Sch}(\mathbf{U}, \Phi)$ whose states are the projections onto $\mathbf{W}$ of the states of $\mathsf{Sch}(\mathbf{U}, \Phi)$ is denoted $\Pi_{\mathbf{W}}^{(\mathbf{U},\Phi)} = (\mathsf{Sch}(\mathbf{W}, \pi_{\mathbf{W}}(\Phi)), \pi_{\mathbf{W}})$. Because this notation can become unwieldy in certain situations, this will be shortened to $\Pi_{\mathbf{U}}^{\mathbf{W}}$ when the constraint set Φ is unambiguously defined by the context.

6.8 The chase inference method Given an $M \in \mathsf{ExtLDB}_\Phi(\mathbf{U})$, there is a natural inference procedure for determining the values of the variables. Informally, the EGD's of Φ are viewed as rules. If the left-hand side of some $\varphi \in \Phi$ is matched by a set of tuples in M under a given substitution α, then the pair of elements on the right-hand side is forced to be equal. If this pair consists of a constant and a variable, then the variable is replaced by the constant. If the right-hand side, under α, consists of two variables, then they are re-named to be the same. If the right hand side, under α, consists of two unequal constants, then M is inconsistent. (This cannot happen if $M \in \mathsf{ExtLDB}_\Phi(\mathbf{U})$.)

This procedure is known as the *chase*, and is sound, complete, and decidable for replacing variables with constants in this way [1, Chap. 10]. For a firm mathematical foundation of how this method is tied to the resolution proof method and more traditional logic, see [10].

In this section, the notation $\Phi \vdash \varepsilon$ will be used to denote that the equality term ε follows from Φ using the chase inference method. The soundness and completeness of this method ensures that this is equivalent to $\Phi \models \varepsilon$.

6.9 Lemma — single-rule inference for equality via EGD's *Let* $M \in \mathsf{ExtLDB}_{\Phi}(\mathbf{U})$*, and let* v_i^A *and* v_j^A *be two variables of the same sort which occur in* M*. Suppose further that* $\Phi \cup M \models (v_i^A =_A v_j^A)$*. Then there is a single* $\varphi \in \Phi^+$ *and a substitution* α *with the following properties.*

(a) $\{\varphi\} \cup M \models (v_i^A =_A v_j^A)$.

(b) *For each antecedent tuple* t_i *of* φ *in the (egd1) form of representation of 6.4,* $\alpha(t_i) \in M$*. Furthermore,* $\alpha(v_i^A) = \alpha(v_j^A)$*.*

In other words, the application of φ *to* M *is via a single direct match of the antecedents of* φ *to the tuples of* M*. The rule* φ *need not be applied repeatedly to obtain the desired equality.*

Proof: The proof is an application to the extant domain of the well-known technique of backward chaining for Horn clauses. Let $\langle \varphi_1, \varphi_2, \ldots \varphi_k \rangle$ be the rules from Φ which are applied, in that order, to effect the inference that $(v_i^A =_A v_j^A)$ holds using the chase inference method. Assume further that each of these rules is in explicit equality form (egd2) of 6.4, and that no two rules have any variables in common. Begin by setting $\varphi := \varphi_k$. For this rule, there are two possibilities for each of its equality antecedents ε_i. First of all, it may be that in the substitution used to instantiate the rule, the equality is already satisfied in M. The other possibility is that the equality is satisfied because some earlier rule φ_j, with $i < j$, forced the two terms to be equal. In the latter case, substitute the set of all antecedents of φ_j for ε_i in φ. The only point of care is that, in this substitution, the variables occurring in the consequent of φ_j must be renamed, throughout the copy of φ_j from which these substituted antecedents are drawn, to match the names in the antecedent ε_i of φ which they replace. This process continues until all antecedent equalities in φ are satisfied by equalities already occurring in M'. Note that as new rules (such as φ_j) are appended to φ in this fashion, new antecedent equalities may appear. However, the process is guaranteed to terminate because replacements are always taken from rules earlier in the proof sequence; new antecedent equalities which are added to φ because a copy of φ_j was appended and which are not satisfied directly in M' can only be satisfied by a substitution of a rule $\varphi_{j'}$ with $j' < j$. This establishes (a).

To show (b), it suffices to note that in the final application of φ, the matches to the antecedent equality atoms are on pairs of terms which are already equal in M. Thus, the matching of φ to M which forced the equality $(v_i^A =_A v_j^A)$ defines the desired α. □

If (N_1, N_2) is a view update to be performed, and M is the current state of the base schema, then the computation of the new state of the base schema proceeds in two steps. First, the $\mathbf{W}$-tuples of N_2 are extended to $\mathbf{U}$-tuples by adding variables in the columns indexed by $\mathbf{U} \setminus \mathbf{W}$. These tuples are then added to M, and the constraints of Φ are used to infer the values of the variables in the added tuples. The formal development of these ideas is as follows.

6.10 Extensions, projective pairs, and completions

(a) Let $t \in \mathsf{Tuples}(\mathbf{W})$. An *extension* of t to $\mathbf{U}$ is any tuple $t' \in \mathsf{Tuples}(\mathbf{U})$ with the property that $t[A] = t'[A]$ for every $A \in \mathbf{W}$. It is a *ground extension* if $t[A] \in \mathsf{Dom}(A)$ for each $A \in \mathbf{U} \setminus \mathbf{W}$, and a *variable extension* if $t[A] \in \mathsf{Vars}(A)$ for each $A \in \mathbf{U} \setminus \mathbf{W}$.

(b) An *extension* of $M \in \mathsf{DB}(\mathbf{W})$ to $\mathbf{U}$ is a set $M' \in \mathsf{DB}(\mathbf{U})$, each tuple of which is an extension of some $t \in M$ to $\mathbf{U}$. Such an extension is a *ground extension* (resp. *variable extension*) if each $t \in M'$ is a ground extension, (resp. variable extension).

(c) A *free extension* of $M \in \mathsf{ExtDB}(\mathbf{W})$ to $\mathbf{U}$ is a variable extension with the further property that no variable in $\mathsf{Vars}(\mathbf{U}) \setminus \mathsf{Vars}(\mathbf{W})$ occurs more than once. Although there are many such free extensions, they are all equivalent up to variable renaming. Therefore, the terminology *the* free extension of M may be used, and a canonical such extension will be denoted by $\mathsf{FreeExt}(M, \mathbf{U})$.

(d) In harmony with 5.1, $(N_1, N_2) \in \mathsf{LDB}_{\pi_{\mathbf{W}}(\Phi)}(\mathbf{W}) \times \mathsf{LDB}_{\pi_{\mathbf{W}}(\Phi)}(\mathbf{W})$ is *upward projective* (resp. *downward projective*, resp. a *strong order pair*) with respect to $\mathsf{Sch}(\mathbf{U}, \Phi)$ if it satisfies the corresponding property of 5.1 with $\Gamma = (\mathbf{V}, \gamma)$ taken to be the view $\Pi_{\mathbf{W}}^{(\mathbf{U},\Phi)} = (\mathsf{Sch}(\mathbf{W}, \pi_{\mathbf{W}}(\Phi)), \pi_{\mathbf{W}})$, and $\mathbf{D}$ the schema $\mathsf{Sch}(\mathbf{U}, \Phi)$.

(e) Let $(N_1, N_2) \in \mathsf{LDB}_{\pi_{\mathbf{W}}(\Phi)}(\mathbf{W}) \times \mathsf{LDB}_{\pi_{\mathbf{W}}(\Phi)}(\mathbf{W})$ be upward projective with respect to $\mathsf{Sch}(\mathbf{U}, \Phi)$, and let $M_1 \in {\pi_{\mathbf{W}}}^{-1}(N_1)$. Define $\mathsf{PreInsert}(M_1, N_2) = M_1 \cup \mathsf{FreeExt}(N_2, \mathbf{U})$. In words, $\mathsf{PreInsert}(M_1, N_2)$ is the extended database M_1' which is obtained from $M_1 \in \mathsf{LDB}_{\pi_{\mathbf{U}}(\Phi)}(\mathbf{U})$ by inserting into it the tuples of $N_1 \in \mathsf{LDB}_{\pi_{\mathbf{W}}(\Phi)}(\mathbf{W})$, suitably extended by variables in the columns of $\mathbf{U} \setminus \mathbf{W}$.

(f) Let $M \in \mathsf{ExtLDB}_{\Phi}(\mathbf{U})$. The *completion* of M, if it exists, is the least $M' \in \mathsf{LDB}_{\Phi}(\mathbf{U})$ with the property that $M \preceq_{\mathbf{U}} M'$. In other words, M' is the completion of M if $M' \leq_{\mathbf{U}} M''$ for any other M'' for which $M \preceq_{\mathbf{U}} M''$. The completion of $M \in \mathsf{ExtLDB}_{\Phi}(\mathbf{U})$ is denoted by $\mathsf{Completion}(M, \Phi)$.

6.11 Proposition *Let* $(N_1, N_2) \in \mathsf{LDB}_{\pi_{\mathbf{W}}(\Phi)}(\mathbf{W}) \times \mathsf{LDB}_{\pi_{\mathbf{W}}(\Phi)}(\mathbf{W})$, *and let* $M \in {\pi_{\mathbf{W}}}^{-1}(N_1)$.

(a) *For each* $t \in N_2 \setminus N_1$, *there is exactly one* $s \in \mathsf{Insert}_{\mathbf{V}}(M, N_2)$ *with the property that* $s[\mathbf{W}] = t$.

(b) *If* (N_1, N_2) *is upward projective, then* $\mathsf{Insert}_{\mathbf{V}}(M, N_2) = \mathsf{Completion}(\mathsf{PreInsert}(M, N_2), \Phi)$.

(c) *If* (N_1, N_2) *is downward projective, then* $\mathsf{Delete}_{\mathbf{V}}(M, N_2) = \{t \in M \mid t[\mathbf{W}] \in N_2\}$.

PROOF: For part (a), if $M' = \mathsf{Insert}_{\mathbf{V}}(M, N_2)$ were to have more than one tuple with the property that $s[\mathbf{U}] = t$, then either one of these tuples could be deleted while retaining the property that $\pi_{\mathbf{W}}(M') = N_2$, which contradicts the infimum condition which defines M'. Part (b) now follows from (a) and the definition of completion. Part (c) follows from 6.6(b); removing tuples cannot result in a violated dependency. □

6.12 The substitution of a completion Continuing with the context above, from 6.11(b) it follows that there must be a substitution α with the property that $\alpha(M) = \mathsf{Completion}(\mathsf{PreInsert}(M, N_2), \Phi)$. This substitution is called the *completing substitution* for M, and is denoted $\mathsf{ComplSub}(M, \Phi)$.

If an update from N_1 to N_2 of the view schema is to be allowed, then it must be allowed for every possible M_1 of the base schema for which $\pi_{\mathbf{W}}(M_1) = N_1$. It is useful to know which constraints on an extension of N_2 to all of $\mathbf{U}$ must hold for every such extension which is the result of an update from N_1 to N_2. This leads to the notion of a most general pre-model.

6.13 Most general extensions Let $N \in \mathsf{LDB}_{\pi_{\mathbf{W}}(\Phi)}(\mathbf{W})$.

(a) A variable extension $M \in \mathsf{ExtLDB}_{\Phi}(\mathbf{U})$ of N is a *strong pre-model* for N if $\alpha(M) \in \mathsf{LDB}_{\Phi}(\mathbf{U})$ for every injective grounding substitution α.

(b) A strong pre-model M for N is a *most-general pre-model* if for any $A \in \mathbf{U} \setminus \mathbf{W}$ and any $t_1, t_2 \in M$, $t_1[A] = t_2[A]$ iff for every $M' \in {\pi_{\mathbf{W}}}^{-1}(N)$ and every $s_1, s_2 \in M'$ with $s_1[\mathbf{W}] = t_1[\mathbf{W}]$ and $s_2[\mathbf{W}] = t_2[\mathbf{W}]$, it is also the case that $s_1[A] = s_2[A]$. A most general pre-model, if it exists, is clearly unique up to a renaming of variables, and so the terminology *the* most-general pre-model will be used.

6.14 Lemma *With conditions as in 6.13 above, the most-general pre-model M is in fact a strong pre-model. That is, $\alpha(M) \in \mathsf{LDB}_{\Phi}(\mathbf{U})$ for every injective grounding substitution α.*

PROOF: Let α be any injective grounding substitution, and let $\varphi \in \Phi$. It is convenient to think of φ as being applied to M in terms of *positions*. A position is specified by a tuple t and an attribute A. Let φ be expressed in explicit equality form (egd2), as defined in 6.4. Suppose that for some substitution β on the variables of φ, the conclusion is that the constants a_i and a_j, which occur in position A of tuples t_1 and t_2, respectively, should be equal. The equalities $\{\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_m\}$ in the antecedent part of the rule are all matched by pairs in positions in the corresponding tuples which must be equal in all models, either because they are positions on attributes in $\mathbf{W}$, in which case they are fixed by N, or else they correspond to attributes in $\mathbf{U} \setminus \mathbf{W}$, in which case these positions are equal in all models, by the definition of the most-general pre-model. Therefore, the variable positions which correspond to a_i and a_j must also be equal in this pre-model, whence a_i and a_j must be the same. Thus, $\alpha(M) \in \mathsf{LDB}_{\Phi}(\mathbf{U})$, as required. $\square$

Notation such as $ABC \to E$ is standard for functional dependencies. The left hand elements, in this case ABC, are the *source* attributes; the values of attributes outside of the source do not affect whether the dependency holds on a given database. Similarly, the attribute E is the *target* attribute; it indicates the value which the dependency can coerce to a certain value. For general EGD's, the situation is a bit more complex, but it is still possible to define the source and target of a dependency; the details follow.

6.15 The source and target of an EGD Assume that the EGD φ is in the explicit equality form (egd2) of 6.4.

(a) The *source* of φ is $\mathsf{Source}(\varphi) = \{\mathsf{Sort}(\varepsilon_i) \mid 1 \leq i \leq m\}$.

(b) The *target* of φ is $\mathsf{Sort}(\varepsilon)$, and is denoted $\mathsf{Target}(\varphi)$.

(c) Let $\mathbf{Y}_1, \mathbf{Y}_2 \subseteq \mathbf{U}$. $\mathsf{EGD}(\mathbf{Y}_1, \mathbf{Y}_2)$ denotes the set of all EGD's φ over $\mathbf{U}$ with the property that $\mathsf{Source}(\varphi) \subseteq \mathbf{Y}_1$ and $\mathsf{Target}(\varphi) \in \mathbf{Y}_2$.

In the context of functional dependencies, If $R[\mathbf{U}]$ decomposes losslessly into $R_{\mathbf{W}}$ and $R_{\mathbf{Y}}$, then there must be a set of FD's of the form $\mathbf{C}_Y \to Y$, one for each $Y \in \mathbf{Y}$, with $\mathbf{C}_Y \subseteq \mathbf{W} \cap \mathbf{Y}$. (Of course, the rôles of $\mathbf{W}$ and $\mathbf{Y}$ could be reversed.) These are known as the *reconstruction dependencies* for the join. A similar situation occurs in the context of EGD's; the corresponding constraints are called *completing dependencies*.

6.16 Completing dependencies

(a) $\varphi \in \Phi^+$ is *internal* to $\mathbf{W}$ if $\mathsf{Source}(\varphi) \cup \{\mathsf{Target}(\varphi)\} \subseteq \mathbf{W}$. The set of all dependencies in Φ^+ which are internal to $\mathbf{W}$ is denoted $\mathsf{IntDep}(\Phi, \mathbf{Y})$.

(b) Let $A \in \mathbf{U} \setminus \mathbf{W}$. The constraint $\varphi \in \Phi^+$ is a *completing dependency* for A from $\mathbf{W}$ if $\varphi \in \mathsf{EGD}(\mathbf{U}, \{A\})$. It is a *source-internal completing dependency* for A from $\mathbf{W}$ if it is in $\varphi \in \mathsf{EGD}(\mathbf{W}, \{A\})$. The set of completing dependencies (resp. source-internal completing dependencies) from $\mathbf{W}$ for some $A \in \mathbf{U} \setminus \mathbf{W}$ is denoted $\mathsf{ComplDep}(\Phi, \mathbf{W}, \mathbf{U})$ (resp. $\mathsf{IntComplDep}(\Phi, \mathbf{W}, \mathbf{U})$).

Completing dependencies which are not source internal are a complicating issue, because the question of whether two values of a target attribute B should be coalesced depends upon the B-values in other tuples. Potentially, this means that whether or not a tuple may be inserted as part of a view update is dependent upon the part of the state of the base schema which is not visible in the view. Clearly, this possibility must be eliminated if a closed update strategy is to be supported. Fortunately, it is always possible to find source-internal dependencies which do the job, as shown next.

6.17 Lemma *Let $A \in \mathbf{U} \setminus \mathbf{W}$, let $(N_1, N_2) \in \mathsf{SPairs}(\Pi_{\mathbf{W}}^{(\mathbf{U},\Phi)})$, and let $M \in {\pi_{\mathbf{W}}}^{-1}(N_1)$. Then for each $v \in \mathsf{Vars}(A)$ which occurs in $\mathsf{PreInsert}(M, N_2)$, there is a $\varphi \in \mathsf{IntComplDep}(\Phi, \mathbf{W}, \mathbf{U})$ with the property that $\{\varphi\} \cup M \models (v =_A \mathsf{ComplSub}(M, \Phi)(v))$. In other words, φ forces v to be bound to the appropriate domain value in the completion.*

PROOF: From 6.9 it follows immediately that there is a $\varphi \in \mathsf{ComplDep}(\Phi, \mathbf{W}, \mathbf{U})$ with the property that $\{\varphi\} \cup M \models (v =_A \mathsf{ComplSub}(M, \Phi)(v))$. It remains to show that φ may be chosen to be in $\mathsf{IntComplDep}(\Phi, \mathbf{W}, \mathbf{U})$. First of all, assume that $M = \alpha(M')$ for some grounding substitution α, with M' the most general pre-model of N_1, as guaranteed by 6.14. Assume further that φ is written in explicit equality form ((egd2) of 6.4). Under a substitution α which matches the antecedents of the rule to M, the only possibility for a match on an attribute $B \in \mathbf{U} \setminus \mathbf{W}$ is on two constants $\{b_0, b_1\}$. However, these two constants can be equal in the most general pre-model only in the case that $\Phi \cup \mathsf{FreeExt}(N_1, \mathbf{U}) \models (\alpha^{-1}(b_0) = \alpha^{-1}(b_1))$. Upon applying 6.9, an EGD φ' is obtained for which $\{\varphi'\} \cup \mathsf{FreeExt}(N_1, \mathbf{U}) \models \{\alpha^{-1}(b_0), \alpha^{-1}(b_1)\}$. In M', this translates to $\{\varphi'\} \cup M' \models (b_0 = b_1)$. Since no two tuples $s_1, s_2 \in \mathsf{FreeExt}(N_1, \mathbf{U})$ have $s_1[B] = s_2[B]$, $\varphi' \in \mathsf{IntComplDep}(\Phi, \mathbf{W}, \mathbf{U})$. Now use backward chaining, as described in the proof of 6.9, to replace the antecedent equality term of φ which matched (b_0, b_1) with the antecedents of rule φ'. After applying this technique to all such pairs from $\mathbf{U} \setminus \mathbf{W}$ which match an antecedent of φ, the resulting rule will be $\mathsf{IntComplDep}(\Phi, \mathbf{W}, \mathbf{U})$, as required.

Finally, this same rule will work for a general M, since the most general pre-model has the least set of matching values possible on the attributes in $\mathbf{U} \setminus \mathbf{W}$, by construction;

having additional equal elements in $\mathbf{U} \setminus \mathbf{W}$ will not cause this rule to fail. $\square$

6.18 Proposition *For each $A \in \mathbf{U} \setminus \mathbf{W}$, there is a subset $\Psi_A \subseteq \Phi^+$ consisting entirely of source-internal completing dependencies for A from $\mathbf{W}$ which may be used to infer the completion of every added tuple in an insertion update. More precisely, for each $(N_1, N_2) \in \mathsf{SPairs}(\Pi_{\mathbf{W}}^{(\mathbf{U},\Phi)})$, each $M \in \pi_{\mathbf{W}}{}^{-1}(N_1)$, and each $v \in \mathsf{Vars}(A) \cap \mathsf{Vars}(\mathsf{PreInsert}(M, N_2))$, $\Psi_A \cup M \models (v =_A \mathsf{ComplSub}(M, \Phi)(v))$.*

PROOF: Just take the union of all of the rules found using the construction of 6.17. Note that this set may not be finite; there is no claim that it is at this point. $\square$

At this point, the focus of this section moves temporarily to a special case, in which the view contains all of the attributes of the base schema, save one. In this context, some very strong results regarding the nature of the completing dependencies, and hence the view updates, may be made. Specifically, it will be concluded that if $\mathsf{SPairs}(\Pi_{\mathbf{W}}^{(\mathbf{U},\Phi)})$ is transitively closed, then the decomposition is governed by a single FD as the completing dependency, and so the family of updates must revert to the familiar, classical case.

6.19 The co-dimension of a view and co-singular views

(a) The *co-dimension* of $\mathbf{W}$ with respect to $\mathbf{U}$ is the number of attributes in $\mathbf{U} \setminus \mathbf{W}$.

(b) $\mathbf{W}$ is said to be *co-singular* to $\mathbf{U}$ if the co-dimension of $\mathbf{W}$ with respect to $\mathbf{U}$ is one. In that case, $\Pi_{\mathbf{W}}^{(\mathbf{U},\Phi)}$ is called a *co-singular* view of $\mathsf{Sch}(\mathbf{U}, \Phi)$.

To establish that the reconstruction dependency is a single FD, it is necessary to rule out all other possibilities. The concept of an EGD being *essentially n-ary* is that any representation of it in the form (egd1) of 6.4 must contain at least three distinct antecedent tuples. In view of 6.5, such a dependency cannot be equivalent to an FD. The formal definitions are given below.

6.20 Representative models and essentially n-ary EGD's Let $\mathbf{W}$ be co-singular to $\mathbf{U}$, with the sole attribute in $\mathbf{U} \setminus \mathbf{W}$ denoted by B. Let $\varphi \in \mathsf{IntComplDep}(\Phi, \mathbf{W}, \mathbf{U})$, and assume that φ is expressed in the (egd1) form of display of 6.4, with the antecedent inequalities expressed within the tuples.

(a) An antecedent tuple t_i of φ is called a *primary tuple* if $t_i[B]$ occurs in the consequent equality ε. Unless φ has a trivial consequent of the form $(v = v)$ (in which case φ is a tautology), there are exactly two primary tuples. All other tuples are called *auxiliary tuples*.

(b) The *extended database of φ*, denoted $\mathsf{ExtDB}(\varphi)$, has as tuples exactly the antecedent tuples of φ. Clearly $\mathsf{ExtDB}(\varphi) \in \mathsf{ExtDB}(\mathsf{Sch}(\mathbf{U}, \pi_{\mathbf{U}}(\boldsymbol{\Phi})))$.

(c) If $\pi_{\mathbf{W}}(\mathsf{ExtDB}(\varphi)) \in \mathsf{ExtLDB}(\mathsf{Sch}(\mathbf{W}, \pi_{\mathbf{W}}(\boldsymbol{\Phi})))$, then φ is called *faithful* for Φ with respect to $\mathbf{W}$.

(d) Assume that φ is faithful. A *representative model* for φ is obtained by applying an injective grounding substitution to $\mathsf{ExtDB}(\varphi)$, and then projecting onto the attributes of $\mathbf{W}$. (In other words, the column for the consequent attribute B is dropped.) It is immediate that such a representative model is in fact in $\mathsf{ExtLDB}(\mathsf{Sch}(\mathbf{W}, \pi_{\mathbf{W}}(\boldsymbol{\Phi})))$, and that all representative models of φ are equivalent up to an injective renaming of the domain values. The *primary* and *auxiliary* tuples of such a model are those which correspond, via the substitution and projection, to the primary and auxiliary tuples of φ.

If M is a representative model for φ, then an *extended representative model* for φ is obtained from M by adding back the column for attribute B, with each tuple in this extension having the same value on B. More precisely, M' is an extended representative model for M if there is $b \in \mathsf{Dom}(B)$, such that $M' = \{t' \in \mathsf{Tuples}(\mathbf{U}) \mid t'[B] = b \text{ and } (\exists t \in M)(t'[\mathbf{W}] = t[\mathbf{W}])\}$. It is clear that M' is a model of φ. The *primary* and *auxiliary* tuples of this extended model M' correspond to those of M. Figure 7 shows an example, which is discussed further in the proof of 6.21.

(e) Let φ be faithful, and let N be a representative model. (In view of the preceding comments, the specific choice of N does not matter.) φ is *essentially n-ary* if it satisfies the following conditions.

  (i) For each primary tuple t, $(N \setminus \{t\}, N) \in \mathsf{SPairs}(\Pi_{\mathbf{W}}^{(\mathbf{U},\Phi)})$.

  (ii) For no primary tuple t and auxiliary tuple t' is it the case that $(N \setminus \{t, t'\}, N \setminus \{t\}) \in \mathsf{SPairs}(\Pi_{\mathbf{W}}^{(\mathbf{U},\Phi)})$.

In other words, either primary tuple t may be deleted from N and then reconstructed using completion, but if any auxiliary tuple is removed, this reconstruction is no longer possible.

The next result is key. It states that if the set of source-completing EGD's for the view contains essentially n-ary constraints, then the set of all strong order pairs for the view cannot be transitively closed, which means in particular that it cannot be a closed update family which is supported by a view update strategy.

6.21 Lemma *Let $\mathbf{W}$ be co-singular to $\mathbf{U}$, with the sole attribute in $\mathbf{U} \setminus \mathbf{W}$ denoted by B. If there is a $\varphi \in \mathsf{IntComplDep}(\Phi, \mathbf{W}, \mathbf{U})$ which is essentially n-ary for some $n > 2$, then $\mathsf{SPairs}(\Pi_{\mathbf{W}}^{(\mathbf{U},\Phi)})$ is not transitively closed.*

PROOF: The proof is most easily illustrated by walking through an example. For this example, let $\mathbf{U} = \{A, B, C, D, E\}$ and let $\mathbf{W} = \{A, B, C, D\}$. In Fig. 7, a rule φ and a corresponding extended representative model M are shown. It is easy to verify that this rule is essentially 4-ary, with the first two tuples primary and the last two auxiliary. This rule is taken to be the only constraint; *i.e.*, $\Phi = \{\varphi\}$. The first two tuples (labelled t_1 and t_2 in the corresponding extended representative model) are primary, while the last two are auxiliary.

The first step is to form the product of M with itself; the result is shown in Fig. 8.

$$\left.\begin{array}{ll} R_{\mathbf{U}}(v_0^A, v_0^B, v_0^C, v_0^D, v_0^E) & \wedge \\ R_{\mathbf{U}}(v_0^A, v_0^B, v_1^C, v_1^D, v_1^E) & \wedge \\ R_{\mathbf{U}}(v_1^A, v_1^B, v_1^C, v_0^D, v_2^E) & \wedge \\ R_{\mathbf{U}}(v_2^A, v_2^B, v_0^C, v_1^D, v_3^E) & \end{array}\right\} \Rightarrow (v_0^E = v_1^E) \qquad \begin{array}{lcl} t_1 & = & (a_0, b_0, c_0, d_0, e_0) \\ t_2 & = & (a_0, b_0, c_1, d_1, e_0) \\ t_3 & = & (a_1, b_1, c_1, d_0, e_0) \\ t_4 & = & (a_2, b_2, c_0, d_1, e_0) \end{array}$$

Figure 7: An EGD and a corresponding extended representative model

$$\begin{array}{lcl}
t_{11} & = & (\langle a_0, a_0\rangle, \langle b_0, b_0\rangle, \langle c_0, c_0\rangle, \langle d_0, d_0\rangle, \langle e_0, e_0\rangle) \\
t_{12} & = & (\langle a_0, a_0\rangle, \langle b_0, b_0\rangle, \langle c_0, c_1\rangle, \langle d_0, d_1\rangle, \langle e_0, e_0\rangle) \\
t_{13} & = & (\langle a_0, a_1\rangle, \langle b_0, b_1\rangle, \langle c_0, c_1\rangle, \langle d_0, d_0\rangle, \langle e_0, e_0\rangle) \\
t_{14} & = & (\langle a_0, a_2\rangle, \langle b_0, b_2\rangle, \langle c_0, c_0\rangle, \langle d_0, d_1\rangle, \langle e_0, e_0\rangle) \\
t_{21} & = & (\langle a_0, a_0\rangle, \langle b_0, b_0\rangle, \langle c_1, c_0\rangle, \langle d_1, d_0\rangle, \langle e_0, e_0\rangle) \\
t_{22} & = & (\langle a_0, a_0\rangle, \langle b_0, b_0\rangle, \langle c_1, c_1\rangle, \langle d_1, d_1\rangle, \langle e_0, e_0\rangle) \\
t_{23} & = & (\langle a_0, a_1\rangle, \langle b_0, b_1\rangle, \langle c_1, c_1\rangle, \langle d_1, d_0\rangle, \langle e_0, e_0\rangle) \\
t_{24} & = & (\langle a_0, a_2\rangle, \langle b_0, b_2\rangle, \langle c_1, c_0\rangle, \langle d_1, d_1\rangle, \langle e_0, e_0\rangle) \\
t_{31} & = & (\langle a_1, a_0\rangle, \langle b_1, b_0\rangle, \langle c_1, c_0\rangle, \langle d_0, d_0\rangle, \langle e_0, e_0\rangle) \\
t_{32} & = & (\langle a_1, a_0\rangle, \langle b_1, b_0\rangle, \langle c_1, c_1\rangle, \langle d_0, d_1\rangle, \langle e_0, e_0\rangle) \\
t_{33} & = & (\langle a_1, a_1\rangle, \langle b_1, b_1\rangle, \langle c_1, c_1\rangle, \langle d_0, d_0\rangle, \langle e_0, e_0\rangle) \\
t_{34} & = & (\langle a_1, a_2\rangle, \langle b_1, b_2\rangle, \langle c_1, c_0\rangle, \langle d_0, d_1\rangle, \langle e_0, e_0\rangle) \\
t_{41} & = & (\langle a_2, a_0\rangle, \langle b_2, b_0\rangle, \langle c_0, c_0\rangle, \langle d_1, d_0\rangle, \langle e_0, e_0\rangle) \\
t_{42} & = & (\langle a_2, a_0\rangle, \langle b_2, b_0\rangle, \langle c_0, c_1\rangle, \langle d_1, d_1\rangle, \langle e_0, e_0\rangle) \\
t_{43} & = & (\langle a_2, a_1\rangle, \langle b_2, b_1\rangle, \langle c_0, c_1\rangle, \langle d_1, d_0\rangle, \langle e_0, e_0\rangle) \\
t_{44} & = & (\langle a_2, a_2\rangle, \langle b_2, b_2\rangle, \langle c_0, c_0\rangle, \langle d_1, d_1\rangle, \langle e_0, e_0\rangle)
\end{array}$$

Figure 8: The product of the instance of Fig. 7

$$\boxed{\begin{array}{cccc} t_{11} & t_{12} & t_{13} & t_{14} \\ t_{21} & t_{22} & t_{23} & t_{24} \\ t_{31} & t_{32} & t_{33} & t_{34} \\ t_{41} & t_{42} & t_{43} & t_{44} \end{array}}$$

For convenience, the tuple name $\langle t_i, t_j\rangle$ is abbreviated to t_{ij}. This product may be arranged in matrix form, as shown to the right. Notice that each row, each column, as well as the diagonal each form a copy of the original model M. Furthermore, these are the only such copies which are embedded in this product. Now, from this product, delete (permanently) all tuples of the form t_{ij} for $i > 2$ and $j > 2$, as well as t_{12}. Call the state at this point M_0. Next, delete t_{22} and call the state M_{21}. Notice that it is still possible to re-insert $\pi_{\mathbf{W}}(t_{22})$ into the view $\Pi_{\mathbf{W}}^{(\mathbf{U},\Phi)}$ and return to M_0, since it is a primary tuple (corresponding to t_2) in the copy of the instantiation of the rule which lies in the second row of the matrix, and all other tuples in this row remain intact. Similarly, if t_{21} is deleted from M_0, resulting in state M_{22}, it is possible to re-insert $\pi_{\mathbf{W}}(t_{21})$ and return to state M_0. In addition, if t_{21} is deleted from state M_{22} to reach state M_1, it is possible to re-insert $\pi_{\mathbf{W}}(t_{21})$ and return to M_{22}. However, it is not possible to re-insert $\pi_{\mathbf{W}}(t_{22})$ directly into M_1, because all copies of the rules on which t_{22} lies have had at least one other tuple removed. This shows that the strong order tuples of the view $\Pi_{\mathbf{W}}^{(\mathbf{U},\Phi)}$ are not transitively closed.

The proof in the general case follows exactly this pattern, and will not be further elaborated. However, there is one further point which should be made. The question might arise as to whether some other strange combination of tuples, not along a row, column, or the diagonal, could represent a dependency which would enable t_{22} to be reinserted. The answer is in the negative. Note that if two tuples $t_{i_1j_1}$ and $t_{i_2j_2}$ in $M \otimes M$ agree on some attribute A, then both projections $\{t_{i_1}, t_{i_2}\}$ and $\{t_{j_2}, t_{j_2}\}$ must agree on those coordinates. Thus, it is impossible to create, in the product, a submodel representing any new pair of tuples which agree on coordinates not already represented by an agreement of two tuples in M. Thus there is no new way to extract a dependency not already embodied in φ. $\square$

As shown below, every *set* of FD's which is not equivalent to a single FD is in fact representable by an essentially n-ary EGD for some $n > 2$. Thus, the only source-completing dependencies which render $\mathsf{SPairs}(\Pi_{\mathbf{W}}^{(\mathbf{U},\Phi)})$ transitively closed are single FD's.

6.22 Lemma *Let* $\mathbf{W}$ *be co-singular to* $\mathbf{U}$*, with the sole attribute in* $\mathbf{U} \setminus \mathbf{W}$ *denoted by* B*, and let* $\mathbf{B}_1, \mathbf{B}_2 \subseteq \mathbf{W}$*. Suppose further that neither* $\mathbf{B}_1 \subseteq \mathbf{B}_2$ *nor* $\mathbf{B}_2 \subseteq \mathbf{B}_1$ *holds. Then there is an essentially ternary EGD* φ *which is logically equivalent to* $\{\mathbf{B}_1 \to B, \mathbf{B}_2 \to B\}$*.*

PROOF: Define the tuples of variables t_1, t_2, and t_3 on attribute set $\mathbf{Y}$ according to the following constraints. $t_1[\mathbf{B}_1] = t_3[\mathbf{B}_1]$, $t_2[\mathbf{B}_2] = t_3[\mathbf{B}_2]$; all other variables are distinct. Let $v_1 = t_1[B]$ and let $v_2 = t_2[B]$, Define $\varepsilon = (v_1 =_B v_2)$, and put $\varphi = (t_1 \wedge t_2 \wedge t_3) \Rightarrow \varepsilon$. It is easy to see that φ is essentially ternary and equivalent to these two functional dependencies. For example, if $\mathbf{Y} = \{A, B, C, D, E\}$ and the two functional dependencies are $AB \to E$ and $BC \to E$, then φ is described by the formula to the right. $\square$

$$\boxed{\left.\begin{array}{lcl} t_1 & = & \langle v_0^A, v_0^B, v_0^C, v_0^D, v_0^E \rangle \\ t_2 & = & \langle v_1^A, v_0^B, v_1^C, v_1^D, v_1^E \rangle \\ t_3 & = & \langle v_0^A, v_0^B, v_1^C, v_2^D, v_2^E \rangle \end{array}\right\} \Rightarrow (v_0^E =_E v_1^E)}$$

Finally, the results are in place to establish the main result for co-singular views — there is a single completing dependency, which must be an FD.

6.23 Theorem — characterization of completing dependencies *Let* $\mathbf{W}$ *be co-singular to* $\mathbf{U}$*, with the sole attribute in* $\mathbf{U} \setminus \mathbf{W}$ *denoted by* B*. The relation* $\mathsf{SPairs}(\Pi_{\mathbf{W}}^{(\mathbf{U},\Phi)})$ *is transitively closed iff there is a* $\mathbf{A} \subseteq \mathbf{W}$ *such that* $(\mathbf{A} \to B) \models \mathsf{IntComplDep}(\Phi, \mathbf{W}, \mathbf{U})$*. In other words, there is a single functional dependency which determines the value of attribute* B *from the attributes of* $\mathbf{W}$ *if and only if the set of strong order pairs of the view* $\Pi_{\mathbf{W}}^{(\mathbf{U},\Phi)}$ *is transitively closed.*

PROOF: If $\mathsf{SPairs}(\Pi_{\mathbf{W}}^{(\mathbf{U},\Phi)})$ is transitively closed, then the property of having a single FD completing dependency follows directly from 6.21 and 6.22 above. On the other hand, if the completing dependency is a single FD $\mathbf{A} \to B$, then a tuple may be inserted iff it matches some existing tuple on the attributes of $\mathbf{A}$, and this is trivially seen to result in an $\mathsf{SPairs}(\Pi_{\mathbf{W}}^{(\mathbf{U},\Phi)})$ which is transitively closed. $\square$

6.24 The reconstruction basis and reconstruction dependencies Let $\mathbf{W}$ be co-singular to $\mathbf{U}$, with the sole attribute in $\mathbf{U} \setminus \mathbf{W}$ denoted by B. Suppose further that $\mathsf{SPairs}(\Pi_{\mathbf{W}}^{(\mathbf{U},\Phi)})$ is transitively closed.

(a) The unique minimal $\mathbf{A} \subseteq \mathbf{W}$ for which $(\mathbf{A} \to B) \models \mathsf{IntComplDep}(\Phi, \mathbf{W}, \mathbf{U})$ is called the *reconstruction basis* for B with respect to Φ, and is denoted $\mathsf{RCBasis}(\mathbf{W}, B, \Phi)$.

(b) The dependency $\mathsf{RCBasis}(\mathbf{W}, A, \Phi) \to B$ itself is called the *reconstruction dependency* for B with respect to Φ, and is denoted $\mathsf{RCDep}(\mathbf{W}, B, \Phi)$.

In terms of decomposition, the above theorem implies that the only possibility is the traditional decomposition based upon the join dependency which is implied by the reconstruction dependency.

6.25 Theorem — characterization of greatest update strategies for co-dimension one *Let* $\mathbf{W}$ *be co-singular to* $\mathbf{U}$*, with the sole attribute in* $\mathbf{U} \setminus \mathbf{W}$ *denoted by* B*. If* $\mathsf{SPairs}(\Pi_{\mathbf{W}}^{(\mathbf{U},\Phi)})$ *is transitively closed, then there is an order-based update strategy for it, in the sense of 3.1, which is given explicitly by* $\mathsf{RawUpdStr}\langle\Pi_{\mathbf{W}}^{(\mathbf{U},\Phi)}, \mathsf{SPairs}(\Pi_{\mathbf{W}}^{(\mathbf{U},\Phi)})\rangle$*. It is precisely the update strategy on* $\mathsf{Sch}(\mathbf{W}, \pi_{\mathbf{W}}(\Phi))$ *which keeps the projection onto the attributes in* $\mathsf{RCBasis}(\mathbf{W}, B, \Phi)$ *constant. It is defined by the complement* $\mathsf{Sch}(\mathbf{C}, \pi_{\mathbf{C}}(\Phi))$*, with* $\mathbf{C} = \mathsf{RCBasis}(\mathbf{W}, B, \Phi) \cup \{B\}$*.*

PROOF: It follows from 6.23 that the minimal EGD φ which reconstructs attribute B is the reconstruction dependency $\mathsf{RCBasis}(\mathbf{W}, B, \Phi) \to B$. From this, it follows that the attributes in $\mathsf{RCBasis}(\mathbf{W}, B, \Phi)$ must be held constant, resulting in the classical relational decomposition based upon a functional dependency defining a join dependency [1, 8.3.3]. $\square$

6.26 Example — schemata with and without universal update families Consider again the example schema $\mathbf{E}_1$, introduced in 1.2. This schema has the single relation $R[ABC]$, with the constraint set $\Phi = \{B \to C\}$. The view to be updated is $\Pi_{AB}^{(\Phi,ABC)} = \Pi_{AB} = (R[AB], \pi_{AB})$. In this case, it is immediate that the reconstruction dependency is $B \to C$, and so the set of strong order pairs of the view Π_{AB} is transitively closed. Therefore, the update family defined by this set of strong order pairs – those updates which hold the projection on attribute B fixed – is a universal update family for Π_{AB}. Thus, there can be no more general order-based update strategy on Π_{AB}; all order-based updates to Π_{AB} must keep the projection Π_B fixed, regardless of the specific strategy.

It is perhaps worth noting explicitly that this result does not do away with the need to check local constraints in the view. For example, let $\mathbf{E}_4$ be as $\mathbf{E}_1$, save that the additional FD $A \to B$ is enforced. Any insertion to Π_{AB} must respect this FD; however, that is a local check within the view, and is in fact embodied within the definition of $\mathsf{SPairs}(\Pi_{AB}^{ABC})$. The point is that if the local constraints are satisfied, then the generation of the completion of the inserted tuple is based upon a single FD, the reconstruction dependency.

The result 6.25 also implies that the example schema $\mathbf{E}_3$ of 4.8 does not admit a universal update family, since the complement attribute C is determined by two distinct FD's, $A \to C$ and $B \to C$. Of course, the two example update strategies presented in 4.8, and their lack of mergeability, already show this, but the above result provides this directly, without the need to search for a counterexample.

Attention is now returned to the general case in which $\mathbf{W}$ is not necessarily co-singular to $\mathbf{U}$. Unfortunately, the results for co-singular views do not extend to the more general case of co-dimension greater than one. To see what the difficulties are, it is appropriate to begin with some further examples.

6.27 Examples — views with co-dimension greater than one Let the schema $\mathbf{E}_5$ be defined by $\mathbf{U} = \{A, B, C, D.E\}$, with $\mathbf{W} = \{A, B, C\}$, and let $\Phi = \{A \to D, B \to E\}$. In this case, the problem of updating the view Π^{ABCDE}_{ABC} is decomposable into the two independent sub-cases of updating Π^{ABCD}_{ABC} and Π^{ABCE}_{ABC}. In the first case, the reconstruction dependency is $A \to D$, while in the latter it is $B \to E$. In other words, $(N_1, N_2) \in \mathsf{SPairs}(\Pi^{ABCDE}_{ABC})$ iff for each $t \in N_2 \setminus N_1$, there are tuples $t_A, t_B \in N_1$ with the properties that $t[A] = t_A[A]$ and $t[B] = t_B[B]$. Thus, upon inserting a new tuple s into N_1, its D coordinate is determined by using the FD $A \to D$, while its E coordinate is determined by using the FD $B \to E$. These two values are determined independently of one another.

Unfortunately, the situation is not always this ideal. Consider the modified schema $\mathbf{E}_6$ in which the FD $D \to E$ is added to Φ. Clearly, the dependencies $A \to D$ and $B \to E$ still govern the reconstruction of attributes D and E, respectively. However, they are no longer sufficient, since the FD $D \to E$ must also remain satisfied after an insertion. Indeed, if $M \in \mathsf{LDB}(\Pi^{ABCDE}_{ABC})$ and $N = \pi_{ABC}(M)$, then the only tuples which may be added to N are those which match an existing tuple on both attributes A and B. In any other case, it is straightforward to construct an $M' \in \mathsf{LDB}(\mathbf{E}_6)$ with $\pi_{ABC}(M) = \pi_{ABC}(M')$, but with the property that FD $D \to E$ is violated upon insertion of the new tuple. Thus, the universal update family in this case is the set of all pairs (N_1, N_2) with the property that $\pi_{AB}(N_1) = \pi_{AB}(N_2)$, and the reconstruction dependencies for D and E are $AB \to D$ and $AB \to E$, respectively.

It is also possible to find a situation in which the set of allowable updates lies properly between these two extremes. Define $\mathbf{E}_7$ by adding, to the second example above, the FD $D \to A$. The total constraint set is now $\Phi = \{A \to D, B \to E, D \to E, D \to A\}$. An intermediate strategy with the flavor of that described in 4.8 is now possible. Specifically, partition the tuples in the schema of the view Π^{ABCDE}_{ABC} into two groups, those which satisfy the FD $A \to B$ and those which satisfy the afunctional dependency $A \not\twoheadrightarrow B$. (See 4.8 for a discussion of afunctional dependencies.) The tuples in the group which satisfy $A \not\twoheadrightarrow B$ may only be updated using the strategy of $\mathbf{E}_6$, that of keeping the AB projection fixed. However, the tuples in the group which satisfies the FD $A \to B$ may be updated using the strategy of $\mathbf{E}_4$, subject to the further requirement that the constraint $A \to B$ be maintained within that group of tuples as well.

Note that, in all of these examples, the FD's $A \to D$ and $B \to E$ define the completion of the update. In other words, whenever an update is allowed, it may be realized by merging the updates for the co-singular views Π^{ABCD}_{ABC} and Π^{ABCE}_{ABC}. The only question is whether further constraints limit which updates are allowed. These observations extend to the general case; the completions are computed column-by-column, and then checked against further constraints for compatibility. Furthermore, if each of the associated co-singular views (in this case Π^{ABCD}_{ABC} and Π^{ABCE}_{ABC}) have a universal update family, then so does the entire view. The details for the general case are now presented.

6.28 Lemma *Let* $\mathbf{Y}$ *be such that* $\mathbf{W} \subseteq \mathbf{Y} \subseteq \mathbf{U}$*, Furthermore, let* ρ *be an update strategy for* Π_W^U*, let* $M_1, M_2 \in \mathsf{LDB}_\Phi(\mathbf{U})$*, with* $\pi_Y(M_1) = \pi_Y(M_2)$*, and let* $N \in \mathsf{LDB}_{\pi_{\mathbf{W}}(\Phi)}(\mathbf{W})$*. Then,* $\rho(M_1, N){\downarrow}$ *iff* $\rho(M_2, N){\downarrow}$*, and when these are defined,* $\pi_Y(\rho(M_1, N)) = \pi_Y(\rho(M_2, N))$*.*

PROOF: The fact that $\rho(M_1, N){\downarrow}$ iff $\rho(M_2, N){\downarrow}$ follows immediately from (upt:1) of 3.1. The equality of the two projections follows from 6.18, which states that each $C \in \mathbf{W}$ has its own set of source-internal completing dependencies with source contained in $\mathbf{W}$. □

6.29 Projection of an update strategy Let $\mathbf{Y}$ be such that $\mathbf{W} \subseteq \mathbf{Y} \subseteq \mathbf{U}$, and ρ be an update strategy for some closed update family T of Π_W^U. Define the partial function $\pi_Y\langle\rho\rangle : \mathsf{LDB}_{\pi_{\mathbf{Y}}(\Phi)}(\mathbf{Y}) \times \mathsf{LDB}_{\pi_{\mathbf{W}}(\Phi)}(\mathbf{W}) \to \mathsf{LDB}_{\pi_{\mathbf{Y}}(\Phi)}(\mathbf{Y})$ as follows. $\pi_Y\langle\rho\rangle(M, N){\downarrow}$ iff $(\pi_W(M), N) \in T$, and then $\pi_Y\langle\rho\rangle(M, N) = \pi_Y(\rho(M', N))$ for any $M' \in \mathsf{LDB}_\Phi(\mathbf{U})$ with $\pi_Y(M') = M$. In view of 6.28, this assignment is independent of the choice of M', and so $\pi_Y\langle\rho\rangle$ is well defined. $\pi_Y\langle\rho\rangle$ is called the *projection of* ρ *onto* $\mathbf{Y}$. It is furthermore an order-based update strategy.

6.30 Lemma — update strategies closed under projection *Let* $\mathbf{Y}$ *be such that* $\mathbf{W} \subseteq \mathbf{Y} \subseteq \mathbf{U}$*, and* ρ *be an update strategy for some closed update family* T *of* Π_W^U*. Then* $\pi_Y\langle\rho\rangle$ *is an update strategy for* T *as well, for the view* Π_W^Y*.*

PROOF: The axioms (upt:1)-(upt:8) of 3.1 are easily established by direct verification. □

Next, the formal ideas of merging update strategies together are developed. Although the notation can be involved at times, the basic idea is quite simple. Since source-internal completing dependencies generate the values for each column in $\mathbf{U} \setminus \mathbf{W}$ separately, it should be possible to glue these separate updates together and then apply more global constraints.

6.31 Merging of databases and update strategies For convenience, let $\mathbf{Z}$ denote $\mathbf{U} \setminus \mathbf{W}$. For each $A \in \mathbf{Z}$, let T_A be a closed update family for the schema $\mathsf{Sch}(\mathbf{W} \cup \{A\}, \pi_{\mathbf{W}\cup\{A\}}(\Phi))$, and let $\rho_A : \mathsf{LDB}_{\pi_{\mathbf{W}\cup\{A\}}(\Phi)}(\mathbf{W} \cup \{A\}) \times \mathsf{LDB}_{\pi_{\mathbf{W}}(\Phi)}(\mathbf{W}) \to \mathsf{LDB}_{\pi_{\mathbf{W}\cup\{A\}}(\Phi)}(\mathbf{W} \cup \{A\})$ be an update strategy for T_A.

(a) For each $A \in \mathbf{Z}$, let $M_A \in \mathsf{LDB}_{\pi_{\mathbf{W}\cup\{A\}}(\Phi)}(\mathbf{W} \cup \{A\})$. $(\bigoplus_{\mathbf{Z}} M_A)$ denotes the set of all $\mathbf{U}$-tuples t which have the property that every projection of the form $\pi_{\mathbf{W}\cup\{A\}}(t)$ lies in M_A. Formally, $(\bigoplus_{\mathbf{Z}} M_A) = \{t \in \mathsf{Tuples}(\mathbf{U}) \mid (\forall A \in \mathbf{Z})(\exists t' \in M_A)(t[\mathbf{W} \cup \{A\}] = t'\}$.

(b) $(\bigoplus_{\mathbf{Z}}^{\mathsf{DB}} \rho_A)$ is the function obtained by pasting together the family of update strategies $\{\rho_A \mid A \in \mathbf{Z}\}$. Formally, define $(\bigoplus_{\mathbf{Z}}^{\mathsf{DB}} \rho_A) : \mathsf{LDB}_\Phi(\mathbf{U}) \times \mathsf{LDB}_{\pi_{\mathbf{W}}(\Phi)}(\mathbf{W}) \to \mathsf{DB}(\mathbf{U})$ by $(M, N) \mapsto (\bigoplus_{\mathbf{Z}} \rho_A(\pi_{\mathbf{W}\cup\{A\}}(M), N))$, with $(\bigoplus_{\mathbf{Z}}^{\mathsf{DB}} \rho_A)(M, N) \downarrow$ iff each $\rho_A(\pi_{\mathbf{W}\cup\{A\}}(M), N){\downarrow}$. In general, $(\bigoplus_{\mathbf{Z}}^{\mathsf{DB}} \rho_A)$ is not an update strategy, because the result is not always in $\mathsf{LDB}_\Phi(\mathbf{U})$.

(c) $(\bigoplus_{\mathbf{Z}} T_A)$ identifies precisely those pairs $(N_1, N_2) \in \mathsf{LDB}_{\pi_{\mathbf{W}}(\Phi)}(\mathbf{W}) \times \mathsf{LDB}_{\pi_{\mathbf{W}}(\Phi)}(\mathbf{W})$ for which $(\bigoplus_{\mathbf{Z}}^{\mathsf{DB}} \rho_A)$ maps into $\mathsf{LDB}_\Phi(\mathbf{U})$, and is called the *merger* of $\{T_A \mid A \in \mathbf{Z}\}$. Formally, define $(\bigoplus_{\mathbf{Z}} T_A) = \{(N_1, N_2) \in \mathsf{LDB}_{\pi_{\mathbf{W}}(\Phi)}(\mathbf{W}) \times \mathsf{LDB}_{\pi_{\mathbf{W}}(\Phi)}(\mathbf{W}) \mid (\forall A \in$

$\mathbf{Z})(\exists(N_1', N_2') \in T_A)((\pi_W(N_1'), \pi_W(N_2')) = (N_1, N_2)) \wedge (\forall M \in \pi_W{}^{-1}(N_1))$ $((\bigoplus_{\mathbf{Z}}^{\mathsf{DB}} \rho_A)(M, N_2) \in \mathsf{LDB}_{\Phi}(\mathbf{U}))\}$.

(d) Restricting $(\bigoplus_{\mathbf{Z}}^{\mathsf{DB}} \rho_A)$ to $(\bigoplus_{\mathbf{Z}} T_A)$ produces an update strategy, by design. Formally, define $(\bigoplus_{\mathbf{Z}} \rho_A) : \mathsf{LDB}_{\Phi}(\mathbf{U}) \times \mathsf{LDB}_{\pi_{\mathbf{W}}(\Phi)}(\mathbf{W}) \to \mathsf{LDB}_{\Phi}(\mathbf{U})$ to be the restriction of $(\bigoplus_{\mathbf{Z}}^{\mathsf{DB}} \rho_A)$ to pairs (M, N) for which $(\pi_{\mathbf{W}}(M), N) \in (\bigoplus_{\mathbf{Z}} T_A)$. $(\bigoplus_{\mathbf{Z}} \rho_A)$ is called the *merger* of $\{\rho_A \mid A \in \mathbf{Z}\}$.

6.32 Theorem — the merger of universal update strategies is universal *Let $\mathbf{Z}$ denote $\mathbf{U} \setminus \mathbf{W}$. For each $A \in \mathbf{Z}$, let T_A be a closed update family for the schema $\mathsf{Sch}(\mathbf{W} \cup \{A\}, \pi_{\mathbf{W}\cup\{A\}}(\Phi))$, and let $\rho_A : \mathsf{LDB}_{\pi_{\mathbf{W}\cup\{A\}}(\Phi)}(\mathbf{W} \cup \{A\}) \times \mathsf{LDB}_{\pi_{\mathbf{W}}(\Phi)}(\mathbf{W}) \to \mathsf{LDB}_{\pi_{\mathbf{W}\cup\{A\}}(\Phi)}(\mathbf{W} \cup \{A\})$ be an update strategy for T_A.*

(a) *$(\bigoplus_{\mathbf{Z}} \rho_A)$ is an order-based update strategy for $(\bigoplus_{\mathbf{Z}} T_A)$.*

(b) *If, for each $A \in \mathbf{Z}$, T_A is a universal update family, then so too is $(\bigoplus_{\mathbf{Z}} T_A)$.*

PROOF: The validity of part (a) is by design; it follows directly from 6.31(c) above. For part (b), it is a straightforward task to verify that each of (upt:1)-(upt:8) of 3.1 is satisfied. To establish universality, suppose that $(N_1, N_2) \in \mathsf{LDB}_{\pi_{\mathbf{W}}(\Phi)}(\mathbf{W}) \times \mathsf{LDB}_{\pi_{\mathbf{W}}(\Phi)}(\mathbf{W})$ is in some closed update family T for $\Pi_{\mathbf{W}}^{(\mathbf{U},\Phi)}$ for which there is an update strategy ρ. In view of 6.30, $(N_1, N_2) \in T_A$ as well, for each $A \in \mathbf{Z}$. However, $(\bigoplus_{\mathbf{Z}} T_A)$ is the largest closed update family for $\Pi_{\mathbf{W}}^{(\mathbf{U},\Phi)}$ which is compatible with each T_A. Thus, it must be that $(N_1, N_2) \in (\bigoplus_{\mathbf{Z}} T_A)$, as required. □

Because of its importance, it is worth restating the major point of 6.32 above explicitly in terms of universality.

6.33 Corollary *Assume that each view of the form $\Pi_{\mathbf{W}}^{\mathbf{W}\cup\{A\}}$ for $A \in \mathbf{U} \setminus \mathbf{W}$ has a universal update family. Then $\Pi_{\mathbf{W}}^{(\mathbf{U},\Phi)}$ also has a universal update family. Furthermore, whenever an update (N_1, N_2) to $\Pi_{\mathbf{W}}^{(\mathbf{U},\Phi)}$ is allowed by this universal update family, it may be realized by merging together the updates to the co-singular views of the form $\Pi_{\mathbf{W}}^{\mathbf{W}\cup\{A\}}$, one for each $A \in \mathbf{U} \setminus \mathbf{W}$.* □

6.34 Liftings in the relational context In Section 5., a number of results were developed for establishing that a set of strong order pairs of a view defines an update strategy. Some of these were not used to establish the results of this section, because more direct approaches are simpler. Nonetheless, the tools of the previous section are useful, and it is instructive to see how concepts such as admitting (5.19(d)) and projecting (5.28) liftings translate to the current framework of single-relation schemata constrained by EGD's.

The ideas are best understood by examining a concrete example. Let $\mathbf{E}_7$ be the schema with the single relation $R[ABCDE]$, constrained by the FD's $AB \to E$ and $B \to D$. Let the view to be updated be Π_{ABCD}. Consider the specific instance M of $\mathbf{E}_7$ shown as the leftmost column of tuples in the display to right. From this, a *generic extended instance* N of the projection onto Π_{ABCD} is constructed. This extended instance is shown as the rightmost column in the display to the right.

$(a_0, b_0, c_0, d_0, e_0)$	(a_0, b_0, v_0^C, d_0)
$(a_0, b_0, c_1, d_0, e_0)$	(a_1, b_0, v_1^C, d_0)
$(a_1, b_0, c_2, d_0, e_1)$	(a_1, b_2, v_2^C, d_2)
$(a_1, b_2, c_0, d_2, e_0)$	(a_2, b_2, v_3^C, d_2)
$(a_2, b_2, c_3, d_2, e_2)$	(a_3, b_3, v_4^C, d_2)
$(a_3, b_3, c_3, d_2, e_3)$	

First of all, the column which does not lie in the view (*i.e.*, E) is deleted. Then certain values are replaced with variables. Specifically, all values which lie in the dependency basis $\{A, B\}$ for the reconstruction dependency $AB \to E$ are left intact. The values for the attribute D are also left intact, since the value for attribute D is determined by the values in the dependency basis, using $B \to D$. The values for attribute C are independent of either FD, and so are replaced with variables. These variables are distinct, even though some of the corresponding values in the instance M are not. They are distinct because there is no constraint which forces them to be otherwise.

This generic extended instance N is easily instantiated to a universal lifting element for any update sequence in which M participates. Specifically, let $\sigma = \langle N_1, N_2, \ldots, N_m \rangle$ be any legal update sequence with $N_1 = \pi_{ABCD}(M)$, and let α be any grounding substitution for which $\alpha(\mathsf{Vars}(N)) \cap \mathsf{Constants}(M) = \emptyset$. Then $\{\{\alpha(N)\}, \{N_1, N_2, \ldots, N_m\}\}$ is easily seen to be a mutual lifting family. Note that N simply encodes the information needed to determine the value of the attribute E from those of the attributes A and B. This information is invariant throughout the update process. In effect, N is a generic instance of those which participate in the orbit of $\pi_{ABCD}(M)$. It is not quite possible to obtain a "universal" lifting for the entire orbit, because every value for attribute D will be used in some instance in the orbit. However, for any finite set of database states which form a legal update sequence, it is always possible to find a grounding substitution α which does not clash with any element in $\mathsf{Dom}(C)$ which is used in an instance of the update sequence. Thus, for this example, the updates admit liftings. This idea is easily extended to any situation in which the reconstruction dependencies are FD's.

It is also easy to see that liftings are projected. In a deletion from M to M', with generic extended instance N for M, to obtain an extended generic instance for M', just delete those tuples from N which no longer correspond to tuples in M'. For example, if (a_1, b_0, c_1, d_0) is deleted from $\pi_{ABCD}(M)$, then (a_1, b_0, v_1^C, d_0) is deleted from N. On the other hand, if (a_0, b_0, c_1, d_0) is deleted from $\pi_{ABCD}(M)$, then no change is made to the generic instance, since equivalent reconstruction information remains in the tuple (a_0, b_0, c_0, d_0). Indeed, these two tuples are in the same orbit, so that (a_0, b_0, c_1, d_0) may be reinserted.

Strong downward transitivity is also satisfied, since deletions correspond to removal of tuples, and this operation is clearly transitive. Finally, the local descending chain condition is trivially satisfied in any relational setting. Thus, the conditions of 5.31 are satisfied for this example.

7. Conclusions and Further Directions

It has been shown that the use of order as a central property of database schemata and morphisms leads to strong uniqueness results for view updates under the constant-complement strategy. Furthermore, techniques for extracting an update strategy directly from a view, without the requirement of first identifying a complement, have been presented. These techniques have been applied to a concrete example – single relation schemata constrained by equality generating dependencies, with projections as views – to illustrate the power of the approach.

Nonetheless, there remains a great deal of work to be done. One issue of interest is that of extending the uniqueness results beyond that of updates which are monotonic under the natural order. As example schema $\mathbf{E}_2$ introduced in 1.4 illustrates, there are common situations in which no admissible update is unique under the natural ordering. In 4.6, it is shown how to remedy this situation for the particular example $\mathbf{E}_2$; however, it remains to develop a general approach. In particular, much work needs to be done to establish the relationship between any such approach and the direct construction techniques developed in Sections 5 and 6.

Another interesting issue is the complexity of updates. The results of Section 6. show that under certain key situations, the reconstruction dependencies are functional dependencies. In view of the known results on constant-time updates for schemata constrained by functional dependencies [21], this means that the complexity of a view update is no more than the complexity of the corresponding update in the main schema. However, for more general settings, additional investigations are in order.

In terms of more practical follow-up work, certainly a closer examination of how these results might be applied to data models other than the relational is high on the list. However, it seems that there is a more fundamental issue which must be addressed first. Most of the research on the view update problem has focused upon open strategies, with the goal to support the largest possible family of updates, even at the expense of violating the isolation principles which are implicit in the constant-complement strategy. Indeed, very little work has been done which builds upon the constant complement strategy, and even in the few cases in which it has, the axioms have been weakened so that critical properties, such as reversibility, have been sacrificed [9]. In the opinion of the author, this situation exists because in traditional database design, the main schema is established first, and only afterwards are views fitted to it. Therefore, *view-centered schema design*, a research direction in which schema design is focused upon the support of views, including view updates based upon the constant-complement strategy, is proposed. The conjecture is that, with the judicious use of null values, most view updates can be accommodated within the constant-complement strategy. The most natural framework in which to pursue these ideas would be within *HERM*, the *H*igher-order *E*ntity-*R*elationship *M*odel [20]; conclusions about many modern data models could be extracted from a study based upon it.

8. Notes Added to the Revised Version

A number of corrections have been made in this on-line version, particularly to 1.1 and 6.31-6.33. Minor typographical corrections have also been made to 1.3, 3.2, 4.7, and 6.20. The author would like to thank Johan Gade for a careful reading which identified these errors.

References

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

[2] J. Adámek, H. Herrlich, and G. Strecker, *Abstract and Concrete Categories*, Wiley-Interscience, 1990.

[3] F. Bancilhon and N. Spyratos, Update semantics of relational views, *ACM Transactions on Database Systems*, **6**(1981), 557–575.

[4] Z. Bellahsène, Updates and object-generating views in ODBS, *Data & Knowledge Engr.*, **34**(2000), 125–163.

[5] F. Bentayeb and D. Laurent, View updates translations in relational databases, in *Proc. DEXA '98, Vienna, Sept. 24-28, 1998*, pp. 322–331, 1998.

[6] B. A. Davey and H. A. Priestly, *Introduction to Lattices and Order*, Cambridge University Press, second edition, 2002.

[7] P. De Bra and J. Paredaens, Horizontal decompositions for handling exceptions to functional dependencies, in *Advances in Data Base Theory, Volume 2* (H. Gallaire, J. Minker, and J. M. Nicolas, eds.), pp. 123–141, Plenum Press, 1984.

[8] R. Fagin, Horn clauses and database dependencies, *JACM*, **29**(1982), 952–985.

[9] G. Gottlob, P. Paolini, and R. Zicari, Properties and update semantics of consistent views, *ACM Trans. Database Systems*, **13**(1988), 486–524.

[10] J. Grant and B. E. Jacobs, On the family of generalized dependency constraints, *JACM*, **29**(1982), 986–997.

[11] G. Grätzer, *Universal Algebra*, D. Van Nostrand, 1968.

[12] S. J. Hegner, Foundations of canonical update support for closed database views, in *ICDT'90, Third International Conference on Database Theory, Paris, France, December 1990* (S. Abiteboul and P. C. Kanellakis, eds.), pp. 422–436, Springer-Verlag, 1990.

[13] S. J. Hegner, Some open problems on view axiomatization, *Bulletin of the EATCS*, (1990), 496–498.

[14] S. J. Hegner, Characterization of desirable properties of general database decompositions, *Ann. Math. Art. Intell.*, **7**(1993), 129–195.

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

[16] S. J. Hegner, Uniqueness of update strategies for database views, in *Foundations of Information and Knowledge Systems: Second International Symposium, FoIKS 2002, Salzau Castle, Germany, February 2002, Proceedings*, pp. 230–249, Springer-Verlag, 2002.

[17] R. Hull, Finitely specifiable implicational dependency families, *Journal of the Association for Computing Machinery*, **31**(1984), 210–226.

[18] E. Mayol and E. Teniente, A survey of current methods for integrity constraint maintenance and view updating, in *Proc. ER '99 Workshops, Paris, Nov. 15-18, 1999*, Springer-Verlag, 1999.

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

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

[21] K. Wang and M. H. Graham, Constant-time maintainability: A generalization of independence, *ACM Trans. Database Systems*, **17**(1992), 201–246.