The Relative Complexity of Updates for a Class of 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

Abstract. It is well known that the complexity of testing the correctness of an arbitrary update to a database view can be far greater than the complexity of testing a corresponding update to the main schema. However, views are generally managed according to some protocol which limits the admissible updates to a subset of all possible changes. The question thus arises as to whether there is a more tractable relationship between these two complexities in the presence of such a protocol. In this paper, this question is answered in the affirmative for closed update strategies, which are based upon the constant-complement approach of Bancilhon and Spyratos. Working within a very general framework which is independent of any particular data model, but which recaptures relational schemata constrained by so-called equality-generating dependencies (EGDs), (which include functional dependencies (FDs)), it is shown that the complexity of testing the correctness of a view update which follows a closed update strategy is no greater than that of testing a corresponding update to the main schema. In particular, if the main schema is relational and constrained by FDs, then there exists a set of FDs on the view, against which any candidate update may be tested for correctness. This holds even though the entire view may not be finitely axiomatizable, much less constrained by FDs alone.

1 Introduction

In a seminal work [1], Bancilhon and Spyratos showed how well-behaved update strategies for database views can be modelled in a very general framework using the so-called constant complement strategy. In more recent work, [2], [3], it is shown that by augmenting this basic framework with natural order structure, true uniqueness for so-called order-based updates may be obtained, in the sense that there is but one way to represent an update to the view in terms of an update to the main schema, regardless of the choice of complement. (*Order-based updates* are those which are realizable as a sequence of insertions and deletions.)

In this paper, the work of [2] and [3] is continued with an initial investigation of the complexity of determining whether a proposed view update is valid. This

is hardly an idle question. Indeed, the axiomatization of a view may be infinitely more complex than that of the base schema, even in very simple cases. For example, the relational schema $\mathbf{E}_1$ with the single relation name $R[ABCD]$ on four attributes and the constraining set of FDs $\mathcal{F}_1 = \{A \rightarrow D, B \rightarrow D, CD \rightarrow A\}$, the projection view Π_{ABC} is not finitely axiomatizable [4].

Even without any special data structures, testing whether a relation satisfies a set of functional dependencies can be performed in time $O(n^2)$, with n the number of tuples in the relation, since it suffices to check each pair of tuples for conflict. If it is known that M is already a legal state and t is a tuple to be inserted, then testing whether $M \cup \{t\}$ satisfies the FDs may be performed in time $O(n)$. Under certain circumstances (e.g., with key dependencies), if the tuples are suitably indexed, these times may be reduced to $O(n)$ and $O(1)$, respectively. On the other hand, for the view Π_{ABC} identified above, neither test can be performed in worst-case $O(n^k)$ for any natural number k. Thus, the increase in complexity is indeed substantial, and certainly dashes any notion of tractability.

However, all is not lost, for testing an arbitrary proposed update to a view for correctness is far more general a task than is testing a proposed update under a closed update strategy. To address the latter idea, a notion of *relative complexity* is introduced, which takes into account that partial information regarding constraint satisfaction is already known about the proposed new view state. To illustrate, let $\mathbf{E}_2$ be the relational schema with the five-attribute relation $S[ABCDE]$, with constraints $\mathcal{F}_2 = \mathcal{F}_1 \cup \{A \rightarrow E\}$. The view to be updated is Π_{ABCE}, with the allowable updates those which hold the complementary view Π_{ABCD} constant. The updates which are allowed under the theory of closed update strategies are precisely those which hold the *meet* of these two views, Π_{ABC}, constant. Now Π_{ABCE} is not finitely axiomatizable, for the same reason that the view Π_{ABC} of $\mathbf{E}_1$ is not. However, since Π_{ABC} is to be held constant under any update to Π_{ABCE}, proposed updates need only be tested against the embedded FD $A \rightarrow E$; it is already known that the "ABC" part of the proposed new database is legal. Thus, the relative complexity of testing a proposed update to Π_{ABCE} is $O(n^2)$, the same as that for proposed updates to the main schema $\mathbf{E}_2$, *even though the view Π_{ABCE} itself is not finitely axiomatizable.* This idea is developed more formally in Example 4.15.

The main result of this paper is that this sort of result holds in a very general context; that is, if the complexity of testing the correctness of a potential database in the main schema is $O(n^k)$, then the relative complexity of testing the correctness of a potential database which is the result of a proposed update under a closed strategy is also $O(n^k)$.

A secondary result is also provided. In [2, 4.2] [3, 4.3], it is shown that the reflection to the main schema of an update to a closed view is unique, provided that the update is realizable as a sequence of legal insertions and deletions. In this paper, it is shown that the intermediate states in fact need not be legal; that it is enough that the update be realizable as sequence of insertions and deletions, and that the initial and final states be legal. In other words, essentially

all updates under closed update strategies in an order-based framework reflect uniquely back to the main view. To illustrate, let $\mathbf{E}_3$ be the schema consisting of the single relation $R[ABC]$, constrained by $\mathcal{F}_3 = \{B \to A, B \to C\}$. The view to be updated is Π_{AB}, with Π_{BC} the complement to be held constant. The legal updates to Π_{AB}, none of which are insertions or deletions, are those which hold Π_B constant and which respect the FD $A \to B$. Thus, the results of [2] [3] do not apply. However, the extensions developed in Sect. 5 do, since the updates may be realized as sequences of insertions and deletions in which the intermediate states may violate the FD $A \to B$. A more detailed explanation is provided in Example 5.9.

2 An Overview of Existing Work

The results presented herein depend heavily upon the earlier work of the author on closed update strategies, which in turn depends upon the initial work of Bancilhon and Spyratos. To provide the reader with the essential background, this section contains two summaries. Summary 2.1 recaps the essential ideas of closed update strategies within the original set-based framework. Thus, it provides the essence of the framework of [1], although it is recast within the formalism of [2] [3]. Summary 2.2 sketches the key ideas developed in [2] [3] which are necessary to extend the set-based ideas to the order-based context.

While every effort has been made to keep this paper self contained, it may nevertheless be necessary to consult [2] and/or [3] to resolve detailed technical issues. Also, while the general theory is not attached to any particular data model, numerous examples are taken from the classical relational theory. Therefore, it is assumed that the reader is familiar with its standard notation and terminology.

Summary 2.1 (The classical results in the set-based framework). In the original work of Bancilhon and Spyratos [1], a database schema $\mathbf{D}$ is just a set. To maintain consistency with the more structured frameworks to be introduced shortly, this set will be denoted $\mathsf{LDB}(\mathbf{D})$ and called the *legal databases* of $\mathbf{D}$. Thus, a database schema is modelled by its instances alone; constraints, schema structure, and the like are not explicitly represented. A *morphism* $f : \mathbf{D}_1 \to \mathbf{D}_2$ of database schemata is a function $f : \mathsf{LDB}(\mathbf{D}_1) \to \mathsf{LDB}(\mathbf{D}_2)$.

A *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 a surjective database morphism. A *morphism* $f : \Gamma_1 \to \Gamma_2$ of views $\Gamma_1 = (\mathbf{V}_1, \gamma_1)$ and $\Gamma_2 = (\mathbf{V}_2, \gamma_2)$ is a morphism $f : \mathbf{V}_1 \to \mathbf{V}_2$ of schemata such that the diagram to the right commutes. Following standard categorical terminology [5, 3.8], the morphism f is an *isomorphism* if there is a $g : \mathbf{V}_2 \to \mathbf{V}_1$ which is both a left and a right inverse to f. The *congruence* of Γ is the equivalence relation on $\mathsf{LDB}(\mathbf{D})$ defined by $(M_1, M_2) \in \mathsf{Congr}(\Gamma)$ iff $\gamma(M_1) = \gamma(M_2)$. It is easy to see that $\Gamma_1 = (\mathbf{V}_1, \gamma_1)$ and $\Gamma_2 = (\mathbf{V}_2, \gamma_2)$ of $\mathbf{D}$ are isomorphic iff $\mathsf{Congr}(\Gamma_1) = \mathsf{Congr}(\Gamma_2)$.

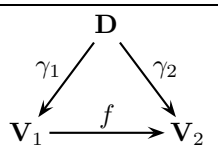

Fig. 1. View morphism

An *update* on the schema $\mathbf{D}$ is a pair $(M_1, M_2) \in \mathsf{LDB}(\mathbf{D}) \times \mathsf{LDB}(\mathbf{D})$, which specifies a change of the state of $\mathbf{D}$ from M_1 to M_2. A *closed update family* U on $\mathbf{D}$ a set of updates which forms an equivalence relation; that is:

(up-r) For each $M \in \mathsf{LDB}(\mathbf{D})$, $(M, M) \in U$.

(up-s) If $(M_1, M_2) \in U$, then $(M_2, M_1) \in U$ as well.

(up-t) If $(M_1, M_2), (M_2, M_3) \in U$, then $(M_1, M_3) \in U$.

Thus, in a closed update family, all updates are reversible, and compatible updates are composable.

Now let $\Gamma = (\mathbf{V}, \gamma)$ be a view of the schema $\mathbf{D}$, and let U and T be closed update families for $\mathbf{D}$ and $\mathbf{V}$ respectively. An update strategy is a rule which translates updates on the view (i.e., in T) to updates on the main schema (i.e., in U). 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 satisfies the following five conditions. (Here $X\downarrow$ means that X is defined.)

(upt:1) $\rho(M, N)\downarrow$ iff $(\gamma(M), N) \in T$. [Admissibility of an update depends only upon the state of $\mathbf{V}$, and not otherwise upon that of $\mathbf{D}$.]

(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.]

The idea of such an update strategy is shown in Fig. 2 below. 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.

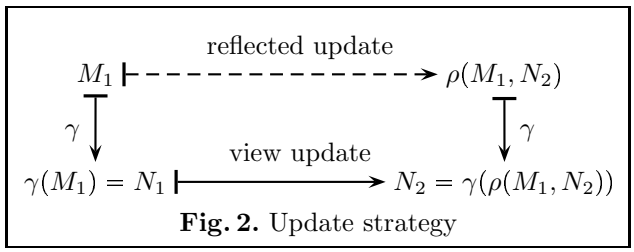

Fig. 2. Update strategy

Some authors have argued that closed update strategies are too restrictive to be of use [6]. However, as shown in [3], they are precisely the view updates which (a) do not depend upon the corresponding state of the main schema for admissibility, and (b) have their effect visible entirely within the view. In other words, they are the updates which can be understood entirely within the context of the view itself.

The idea that all view updates in a closed strategy have their effect contained entirely within the view itself is further manifested in their characterization via constant complement. A pair $\{\Gamma_1 = (\mathbf{V}_1, \gamma_1), \Gamma_2 = (\mathbf{V}_2, \gamma_2)\}$ of views of the schema $\mathbf{D}$ is called a subdirect complementary pair if it defines a lossless decomposition of $\mathbf{D}$. More precisely, the product $\Gamma_1 \times \Gamma_2 = (\mathbf{V}_1 \,{}_{\gamma_1}\!\otimes_{\gamma_2} \mathbf{V}_2, \gamma_1 \otimes \gamma_2)$

has $\mathsf{LDB}(\mathbf{V}_1 {}_{\gamma_1}\!\otimes_{\gamma_2}\mathbf{V}_2) = \{(\gamma_1(M), \gamma_2(M)) \mid M \in \mathsf{LDB}(\mathbf{D})\}$. 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 $M \mapsto (\gamma_1(M), \gamma_2(M))$. The pair $\{\Gamma_1, \Gamma_2\}$ is said to form a *subdirect complementary pair*, and Γ_1 and Γ_2 are called *subdirect complements* of one another, just in case $\gamma_1 \otimes \gamma_2$ is a bijection. In other words, $\{\Gamma_1, \Gamma_2\}$ is a subdirect complementary pair precisely in the case that the state of the schema $\mathbf{D}$ is recoverable from the combined states of $\mathbf{V}_1$ and $\mathbf{V}_2$. In the classical relational setting, this amounts to a lossless decomposition.

If $\{\Gamma_1, \Gamma_2\}$ is a subdirect complementary pair, it is clear that there can be at most one update strategy on Γ_1 which holds Γ_2 constant. Specifically, 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 $(M, N) \mapsto (\gamma_1 \otimes \gamma_2)^{-1}(N, \gamma_2(M))$ whenever $(N, \gamma_2(M)) \in \gamma_1 \otimes \gamma_2(\mathsf{LDB}(\mathbf{D}))$, and undefined otherwise. $\mathsf{UpdStr}\langle\Gamma_1, \Gamma_2\rangle$ is called the *update strategy* for Γ_1 with constant complement Γ_2. As first shown by Bancilhon and Spyratos [1, Thm. 7.3], every closed update strategy on a view Γ is of the form $\mathsf{UpdStr}\langle\Gamma, \Gamma'\rangle$ for some view Γ'. Specifically, let T and U closed update strategies for $\mathbf{V}$ and $\mathbf{D}$ respectively, and ρ an update strategy for T with respect to U. 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 $\{(M_1, M_2) \in \mathsf{LDB}(\mathbf{D}) \mid (\exists N \in \mathsf{LDB}(\mathbf{V}))(\rho(M_1, N) = M_2)\}$. The *$\rho$-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 $M \mapsto [M]_{\equiv_\rho}$, with the latter denoting the equivalence class of M in $\equiv_\rho$. In other words, $(M_1, M_2) \in \mathsf{Congr}(\tilde{\Gamma}^\rho)$ iff some view update $(N_1, N_2) \in T$ changes the state of $\mathbf{D}$ from $M_1 \in \gamma^{-1}(N_1)$ to $M_2 \in \gamma^{-1}(N_2)$. Thus, by construction, a potential update $(N_1, N_2) \in \mathsf{LDB}(\mathbf{V})$ is supported under ρ iff for some (resp. any) $M_1 \in \gamma^{-1}(N_1)$, there is an $M_2 \in \mathsf{LDB}(\mathbf{D})$ with $(M_1, M_2) \in \mathsf{Congr}(\tilde{\Gamma}^\rho)$. In other words, the allowable updates to Γ under ρ are precisely those whose reflection into $\mathbf{D}$ leaves $\tilde{\mathbf{V}}^\Gamma$ fixed; i.e., $\rho = \mathsf{UpdStr}\langle\Gamma, \tilde{\Gamma}^\rho\rangle$.

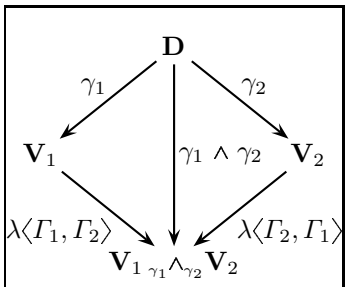

Fig. 3. Relative views

Not all subdirect complements give rise to closed update strategies. Condition (upt:1) mandates that the admissibility of a view update depend upon the state of the view alone. Thus, any information which is contained in the complement view and which is needed to determine the admissibility of an update must be contained in the view to updated as well. The necessary condition, first observed in [7, 2.10], is that the congruences must commute. Formally, 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. In this case, $\mathsf{Congr}(\Gamma_1) \circ \mathsf{Congr}(\Gamma_2)$ is also an equivalence relation on $\mathsf{LDB}(\mathbf{D})$, and so it is possible to define (up to isomorphism) the view $\Gamma_1 \wedge \Gamma_2 = (\mathbf{V}_1 {}_{\gamma_1}\!\wedge_{\gamma_2}\mathbf{V}_2, \gamma_1 \wedge \gamma_2)$ with $\mathsf{Congr}(\Gamma_1 \wedge \Gamma_2) = \mathsf{Congr}(\Gamma_1) \circ \mathsf{Congr}(\Gamma_2)$. The situation is summed up in the

Fig. 3 above. Note that $\mathbf{V}_1 \,{}_{\gamma_1}\!\wedge_{\gamma_2}\mathbf{V}_2$ is a view not only of $\mathbf{D}$, but of $\mathbf{V}_1$ and $\mathbf{V}_2$ as well. Now 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 a closed update family, called the *update family* induced by Γ_2 on Γ_1. The key result [2, 3.9] [3, 3.10] is the following:

(a) For any update strategy ρ, $\mathsf{UpdStr}\langle\Gamma,\tilde{\Gamma}^{\rho}\rangle = \rho$.
(b) For any meet complement Γ_1 of Γ, $\tilde{\Gamma}^{\mathsf{UpdStr}\langle\Gamma,\Gamma_1\rangle} = \Gamma_1$.

In the context of relational schema and views defined by projection, a pair of views forms a meet-complementary pair iff the decomposition is both lossless and dependency preserving [2, 2.16] [3, 2.17]. In this case, the meet view is just the projection on the common columns. To obtain an example in which the views form a subdirect complementary pair but not a meet complementary pair, it suffices to consider an example which is lossless but not dependency preserving.

In [8], the connection between decompositions of database schemata and commuting congruences is investigated thoroughly.

Summary 2.2 (The order-based framework). Despite its simplicity and elegance, the set-based framework for closed update strategies has a substantial shortcoming; namely, the update strategy depends upon the choice of the complement. The theory cannot distinguish between complements, even those which yield identical meets, and so identical update families. For example, let $\mathbf{E}_1$ be the relational schema with the single relation $R[ABC]$, constrained by the single FD $B \to C$, and let Π_{AB} be the view defined by the projection mapping π_{AB}. Define Π_{BC} similarly. Since the pair $\{\Pi_{AB},\Pi_{BC}\}$ forms a lossless and dependency-preserving decomposition of $\mathbf{E}_1$, it also forms a meet-complementary pair [2, 2.16] [3, 2.17]. Indeed, Π_{BC} is the "natural" complement of Π_{AB}, and the one which yields the "obvious" strategy for reflecting updates to Π_{AB} back to $\mathbf{F}_0$. However, as shown in [2, 1.3] [3, 1.3], it is possible to find other complements of Π_{AB} which have exactly the same meet, and so support exactly the same updates to Π_{AB}. Although these alternate complements are a bit pathological, the set-based theory outlined above in Summary 2.1 does not prefer Π_{BC} to them in any way.

To formalize this preference, additional structure must be incorporated into the model. Most database models incorporate some sort of order structure. In the relational model, the databases may be ordered via relation-by-relation inclusion. Furthermore, the common database mappings built from projection, restriction, and join are all order preserving with respect to this natural order structure. In particular, while the views Π_{AB} and Π_{BC} are order mappings, the alternate views identified in [2, 2.16] [3, 2.17] are not.

The theory developed in [2] and [3] provides a systematic extension to the results outlined in Summary 2.1 above to the order-based setting. A *order schema* $\mathbf{D}$ is taken to be a partially ordered set (*poset*) $(\mathsf{LDB}(\mathbf{D}),\leqslant_{\mathbf{D}})$. A *order database mapping* $f : \mathbf{D}_1 \to \mathbf{D}_2$ is an order-preserving function; i.e., $M_1 \leqslant_{\mathbf{D}_1} M_2$ implies $f(M_1) \leqslant_{\mathbf{D}_2} f(M_2)$. An *order view* $\Gamma = (\mathbf{V},\gamma)$ of $\mathbf{D}$ consists of an order schema $\mathbf{V}$ and an open surjection $\gamma : \mathsf{LDB}(\mathbf{D}) \to \mathsf{LDB}(\mathbf{V})$; that is, a surjection which is order preserving and, in addition, which satisfies the property that whenever

$N_1 \leqslant_{\mathbf{V}} N_2$, there are $M_1, M_2 \in \mathsf{LDB}(\mathbf{D})$ with the properties that $M_1 \leqslant_{\mathbf{D}} M_2$, $f(M_1) = N_1$, and $f(M_2) = N_2$. For the pair of order views $\{\Gamma_1, \Gamma_2\}$ to be a subdirect complementary pair in the order sense, the mapping $\gamma_1 \otimes \gamma_2 : \mathbf{D} \to \mathbf{V}_1 {}_{\gamma_1}\!\otimes_{\gamma_2} \mathbf{V}_2$ must be an order isomorphism, and not merely an order-preserving bijection. To obtain a closed update strategy in the oder-bases sense, conditions (upt:1)-(upt:5) identified in Summary 2.1 are augmented with the following three additions.

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

(upt:7) If $\rho(M_1, N_1)\downarrow$ with $M_1 \leqslant_{\mathbf{D}} \rho(M_1, N_1)$, then for all $M_2 \in \mathsf{LDB}(\mathbf{D})$ with $M_1 \leqslant_{\mathbf{D}} M_2 \leqslant_{\mathbf{D}} \rho(M_1, N_1)$, 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 $M_1 \leqslant_{\mathbf{D}} M_2$, then for every $N_1, N_2 \in \mathsf{LDB}(\mathbf{V})$ for which $N_1 \leqslant_{\mathbf{V}} N_2$, $\rho(M_1, N_1)\downarrow$, and $\rho(M_2, N_2)\downarrow$, if must be the case that $\rho(M_1, N_1) \leqslant_{\mathbf{D}} \rho(M_2, N_2)$. [This condition is called *order reflection*.]

Modulo these modification, it is fair to say, at least in a general way, that [2] and [3] extend the classical set-based constant complement theory to the order-based setting. Within the setting of this extension, a number of uniqueness results are obtained. Most importantly, while complements are not necessarily unique, order-based updates are. Specifically, let $\mathbf{D}$ be a database schema, and let U be a closed update family for $\mathbf{D}$. A pair $(M_1, M_2) \in U$ is called: a *formal insertion* with respect to U if $M_1 \leqslant_{\mathbf{D}} M_2$; a *formal deletion* with respect to U if $M_2 \leqslant_{\mathbf{D}} M_1$; and an *order-based update* with respect to U if it is a composition of a sequence of formal insertions and formal deletions. The main theorem [2, 4.2] [3, 4.3] states that for an order-based view $\Gamma = (\mathbf{V}, \gamma)$ of the order-based schema $\mathbf{D}$, all order-based closed update strategies must agree on all order-based updates. In other words, there is only one way to reflect the view update back to the main schema. This does not depend upon the choice of complement, or even the value of the meet. It is unique, period. As a rich source of classical but important examples, all SPJR-mappings (Select, Project, Join, Rename) in the classical relational setting define order views [2, 2.5] [3, 2.5].

In [9], a theory of *direct* decomposition (i.e., situations in which the views are independent and so the meet is trivial) of order-based schemata is presented.

3 A Framework for Modelling View Updates

The framework described in Summary 2.2 must be extended in two essential ways in order to recapture the key ideas involved in updates and their complexity. First of all, to recapture complexity, it must be possible to characterize the size of a database, and also the size of an update. Secondly, to recapture admissibility of a candidate database, it must be possible to discuss both those databases which satisfy the underlying constraints and those which do not. Fortunately, there is a very simple model which meets both of these requirements. To begin, the underlying ideas in the world of posets are developed.

Definition 3.1 (CFA-posets and morphisms). Let X be any set (not necessarily finite), and let $\boldsymbol{\mathscr{P}}_f(X) = (\mathscr{P}_f(X), \subseteq)$ denote the poset consisting of all finite subsets of X, ordered under set inclusion. A *concrete finitely-atomistic poset* $\mathbf{P} = (P, \subseteq)$ *(over X)* (*CFA-poset* for short) is any sub-poset of $\boldsymbol{\mathscr{P}}_f(X)$ which contains the least element $\varnothing$, as well as all singletons of the form $\{x\}$ with $x \in X$. Define $\mathsf{Atoms}(\mathbf{P}) = \{\{x\} \mid x \in X\}$; these are clearly the atoms of this poset in the abstract sense [10, 5.2]. The *basis* of any $p \in P$ is $\mathsf{Basis}_{\mathbf{P}}(p) = \{a \in \mathsf{Atoms}(\mathbf{P}) \mid a \subseteq p\} = \{x \in X \mid x \in p\}$. The term *finitely atomistic* is borrowed from the lattice-theoretic world [11, p. 234], and refers to the fact that every element in $\mathbf{P}$ is the supremum of the atoms which are less than it; i.e., $p = \sup\{a \in \mathsf{Atoms}(\mathbf{P}) \mid a \subseteq p\}$. Note also that X may be recovered from $\mathbf{P}$ as $\bigcup \mathsf{Atoms}(\mathbf{P}) = \{x \mid \{x\} \in \mathsf{Atoms}(\mathbf{P})\}$, so that it is safe to speak of a CFA-poset without explicitly identifying the underlying set.

To avoid confusion when more than one poset is considered, $\bot_{\mathbf{P}}$ will be used to denote $\varnothing$ when it is regarded as the least element of $\mathbf{P}$. Finally, it is often useful to have a notation for atoms and basis when the least element is included as well; thus $\mathsf{ExtAtoms}(\mathbf{P}) = \mathsf{Atoms}(\mathbf{P}) \cup \{\bot_{\mathbf{P}}\}$, and for $p \in P$, $\mathsf{ExtBasis}_{\mathbf{P}}(p) = \mathsf{Basis}_{\mathbf{P}}(p) \cup \{\bot_{\mathbf{P}}\}$,

Let $\mathbf{P} = (\mathsf{LDB}(\mathbf{P}), \subseteq)$ and $\mathbf{Q} = (\mathsf{LDB}(\mathbf{Q}), \subseteq)$ be CFA-posets. A *CFA-morphism* is a function $f : P \to Q$ with the property that it is *basis preserving*, in the precise sense that for all $p \in P$, $\bigcup\{f(a) \mid a \in \mathsf{Basis}_{\mathbf{P}}(p)\} = \mathsf{Basis}_{\mathbf{Q}}(f(p))$. It is clear that a CFA-morphism is *monotone*, i.e., $p_1 \subseteq p_2$ implies $f(p_1) \subseteq f(p_2)$, and thus a poset morphism in the ordinary sense. Furthermore, $f(\bot_{\mathbf{P}}) = \bot_{\mathbf{Q}}$, since $\mathsf{Basis}_{\mathbf{P}}(\bot_{\mathbf{P}}) = \mathsf{Basis}_{\mathbf{Q}}(\bot_{\mathbf{Q}}) = \varnothing$. Thus, the behavior of a basis-preserving morphism is determined entirely by its action on the atoms of the poset.

The CFA-morphism $f : \mathbf{P} \to \mathbf{Q}$ is *open* if, for each pair $q_1, q_2 \in Q$ with $q_1 \subseteq q_2$, there are $p_1, p_2 \in P$ with the properties that $p_1 \subseteq p_2$, $f(p_1) = q_1$, and $f(p_2) = q_2$. A CFA-morphism which is surjective is called a *CFA-surjection*. If f is both open and surjective, then $\mathbf{Q}$ carries the weakest order which renders f monotone.

Definition 3.2 (CFA-schemata, morphisms, and views). Formally, a *concrete finitely atomistic database schema* (*CFA-schema* for short) $\mathbf{D} = (\mathsf{LDB}(\mathbf{D}), \subseteq)$ is just a CFA-poset. A *CFA-database morphism* is just a CFA-morphism in the sense given above. A *CFA-view* is a pair $\Gamma = (\mathbf{V}, \gamma)$ in which $\mathbf{V}$ is a CFA-schema and γ is an open CFA-surjection.

Example 3.3 (Relational CFA-schemata, morphisms, and views). Let $\mathbf{R}$ be a relational schema consisting of a single relation $R[\mathbf{A}]$, with a family $\mathcal{C}$ of constraints; $\mathsf{LDB}(\mathbf{R})$ the set of all finite relations satisfying those constraints. $\mathbf{R}$ is automatically an order schema in the sense of [2] and [3], with the order defined by set inclusion. For it to be a CFA-schema, it must also be finitely atomistic. Specifically, this means that both the empty relation $\varnothing$ and each set $\{t\}$ containing exactly one tuple satisfies the constraints of $\mathcal{C}$. These conditions are satisfied, for example, whenever $\mathcal{C}$ consists of universal dependencies, such as *full dependencies* [12, Ch. 10].) These sets are very broad, and include *equality*

generating dependencies (EGDs) such as FDs, and *tuple generating dependencies* (TGDs) such as join dependencies. They do not, however, include dependencies involving existential quantification, such as *inclusion dependencies* [12, Ch. 9], upon which foreign-key dependencies are based. A similar construction applies in the case in which $\mathbf{R}$ contains several relations; in this case, the atoms are those instances in which one relation contains one tuple, and the rest are empty.

Both projection and restriction are examples of open CFA-morphisms in the relational context, since each is defined by its action on single tuples. Thus, they define CFA-views. On the other hand, joins are not in general CFA-morphisms, since they are not basis preserving.

Definition 3.4 (Unconstrained databases). While the legal databases consist of some finite subsets of the foundation, the potential databases consist of all of them. Specifically, Let $\mathbf{D} = (\mathsf{LDB}(\mathbf{D}), \subseteq)$ be a CFA-schema, and define $(\overline{\mathbf{D}} = \mathsf{DB}(\mathbf{D}), \subseteq)$ with $\mathsf{DB}(\mathbf{D}) = \mathscr{P}_f(\mathsf{Foundation}(\mathbf{D}))$, and let $\iota_{\mathbf{D}} : \mathbf{D} \to \overline{\mathbf{D}}$ be the identity embedding; $M \mapsto M$. The elements of $\mathsf{DB}(\mathbf{D})$ are called the *unconstrained databases* of $\mathbf{D}$. In the relational example $\mathbf{R}$ identified in Example 3.3, $\mathsf{DB}(\mathbf{D})$ is the set of all finite relations on attribute set $\mathbf{A}$, regardless of whether or not the constraints of $\mathcal{C}$ are met.

Remark 3.5 (Abstract FA-schemata). A cornerstone of the framework developed in [2] [3] is that equivalence up to poset isomorphism is adequate to characterize a database schema. In other words, the theory is indifferent to such "inessential" variations. However, in the approach taken here, this is not the case. Rather, the database schemata are required to have a specific form; namely, that of a sub-poset of a power set. This is not an essential change. It is quite possible to develop the theory of this paper along the lines of *abstract FA-posets*, which may be axiomatized independently of any reference to CFA-posets, but which amount to those posets which are isomorphic to CFA-posets. The reason for not taking this direction is that it becomes much less intuitive, and much more cumbersome notationally, to define the unconstrained databases. The gains realized in having such a natural model for going from constrained to unconstrained databases seems worth the loss in abstraction. The more concrete approach is not entirely without its drawbacks, however. In particular, views which are constructed axiomatically must then be "concretized." For example, at the end of Definition 4.3, a method for constructing a CFA-view from a congruence is provided.

Definition 3.6 (k-models and subinstance properties). In a relational schema $\mathbf{R}$ constrained by a family $\mathcal{F}$ of FDs, to test a candidate relation M for legality, it suffices to test each pair of tuples for conflict. In other words, if every two-element subset of M satisfies $\mathcal{F}$, then M itself does. For more general families of EGDs, a corresponding property requiring the testing of k tuples at a time is easily formulated. The following notions extend these ideas to the abstract framework.

Let $\mathbf{D} = (\mathsf{LDB}(\mathbf{D}), \subseteq)$ be a CFA-database schema, and let $k \in \mathbb{N}$. A *k-model* of $\mathbf{D}$ is any $M \in \mathsf{DB}(\mathbf{D})$ with the property that every $N \in \mathsf{DB}(\mathbf{D})$ with $N \subseteq M$ and

$\mathsf{Card}(\mathsf{Basis}_{\overline{\mathbf{D}}}(N)) \leqslant k$ is in $\mathsf{LDB}(\mathbf{D})$. (Here $\mathsf{Card}(X)$ denotes the cardinality of the set X.) The schema $\mathbf{D}$ has the *k-submodel property* if, for every $M \in \mathsf{DB}(\mathbf{D})$, $M \in \mathsf{LDB}(\mathbf{D})$ iff it is a k-model of $\mathbf{D}$. The schema $\mathbf{D}$ is *closed under subinstances* if, for any $M \in \mathsf{LDB}(\mathbf{D})$ and any $N \subseteq \mathsf{Basis}_{\mathbf{D}}(M)$, $N \in \mathsf{LDB}(\mathbf{D})$ as well. The following observation is immediate.

Observation 3.7. *If $\mathbf{D}$ is a CFA-schema which has the k-submodel property for some natural number k, then it is closed under subinstances.* □

Remark 3.8 (The limits of k-models). While k-models recapture the idea of FDs in particular and EGDs in general, they do not recapture the tuple-generating properties of join dependencies in particular and TGDs in general. For such dependencies, a more general notion, the (k_1, k_2)-model, is needed. See the comments in Discussion 6.1.

Remark 3.9 (The measure of complexity). In a simple sense, a potential database $M \in \overline{\mathbf{D}}$ of a schema $\mathbf{D}$ with the k-submodel property may be tested for membership in $\mathsf{LDB}(\mathbf{D})$ in worst-case time $O(n^k)$, with $n = \mathsf{Card}(\mathsf{Basis}_{\mathbf{D}}(M))$, since there are $\binom{n}{k}$ subsets of size k to test, and $O(\binom{n}{k}) = O(n^k)$, when k is taken to be constant and n the variable. In the context of simple update operations, this complexity may be further reduced. For example, if the update corresponds to the insertion of a single atom, then the complexity is $O(n^{k-1})$, since the only k-element subsets which need be checked are those whose basis contains the new atom. In view of Observation 3.7, no checking is needed at all for deletions.

In certain contexts, with the support of appropriate data structures, these values may be reduced even further. Most notably, with key dependencies in the case of FDs, satisfaction may be performed in linear time, and the correctness of simple insertions may be determined in constant time [13]. For reasons of this dependence upon data structures, as well as space constraints, these issues will not be pursued further in this paper. Rather, complexity will be characterized solely in terms of k-submodel properties.

To close this section, a few essential properties of schemata which are closed under subinstances are developed.

Definition 3.10 (Strong morphisms and injective generators). Let $\mathbf{D}_1 = (\mathsf{LDB}(\mathbf{D}_1), \subseteq)$ and $\mathbf{D}_2 = (\mathsf{LDB}(\mathbf{D}_2), \subseteq)$ be CFA-schemata, and let $f : \mathbf{D}_1 \to \mathbf{D}_2$ be a CFA-surjection.

(a) The morphism f is a *downwardly strong* if for every $M \in \mathsf{LDB}(\mathbf{D}_1)$ and $N \in \mathsf{LDB}(\mathbf{D}_2)$ for which $N \subseteq f(M)$, there is an $M' \in \mathsf{LDB}(\mathbf{D}_1)$ with $M' \subseteq M$ and $f(M') = N$. Note in particular that if f is surjective and downwardly strong, then it must be open as well.

(b) The morphism f is *injectively generating* if for every $M_2 \in \mathsf{LDB}(\mathbf{D}_2)$, there is an $M_1 \in f^{-1}(M_2)$ with the property that $\mathsf{Card}(\mathsf{Basis}_{\mathbf{D}_1}(M_1)) = \mathsf{Card}(\mathsf{Basis}_{\mathbf{D}_2}(M_2))$.

Proposition 3.11. *Let $\mathbf{D}_1 = (\mathsf{LDB}(\mathbf{D}_1), \subseteq)$ and $\mathbf{D}_2 = (\mathsf{LDB}(\mathbf{D}_2), \subseteq)$ be a CFA-database schemata, with $f : \mathbf{D}_1 \to \mathbf{D}_2$ a CFA-surjection and $\mathbf{D}_1$ closed under*

subinstances. Then f is downwardly strong and injectively generating, and $\mathbf{D}_2$ is closed under subinstances.

Proof. To show that f is downwardly strong, let $M \in \mathsf{LDB}(\mathbf{D}_1)$ and $N \in \mathsf{LDB}(\mathbf{D}_2)$ with $f(M) = N$, and let $N' \in \mathsf{LDB}(\mathbf{D}_2)$ with $N' \subseteq N$. Set $M' = \{A \in M \mid f(A) \in N'\}$. Then $M' \in \mathsf{LDB}(\mathbf{D}_1)$ (since $\mathbf{D}_1$ is closed under subinstances) with $f(M') = N'$.

To show that f is injectively generating, let $N \in \mathsf{LDB}(\mathbf{D}_2)$ and choose $M \in f^{-1}(N)$. For each $B \in N$, choose exactly one $A_B \in M$ with the property that $f(A) = B$, and put $M' = \{A_B \mid B \in N\}$. Since $\mathbf{D}_1$ is closed under subinstances, $M' \in \mathsf{LDB}(\mathbf{D}_1)$, with $f(M') = N'$.

Finally, to show that $\mathbf{D}_2$ is closed under subinstances, let $N \in \mathsf{LDB}(\mathbf{D}_2)$ and let $N' \in \mathsf{DB}(\mathbf{D})$ with $N' \subseteq N$. Choose $M \in f^{-1}(N)$ and set $M' = \{A \in M \mid f(A) \in N'\}$. Since $\mathbf{D}_1$ is closed under subinstances, $M' \in \mathsf{LDB}(\mathbf{D}_1)$, and so $N' \in \mathsf{LDB}(\mathbf{D}_2)$ with $f(M') = N'$. □

4 View Constructions and Relative Complexity

The ultimate goal of this section is the proof of the main theorem of this paper — that the relative complexity of view update for a closed view is no greater than that of update in the base schema. To achieve this result, certain key results from [2] [3] must be lifted to the current, more structured framework.

To begin, it is shown that a functional connection, or, equivalently, a subsumption of congruences is sufficient to define a relative view.

Lemma 4.1 (CFA-view fill-in). *Let $\mathbf{D} = (\mathsf{LDB}(\mathbf{D}), \subseteq)$ be a CFA-schema, let $\Gamma_1 = (\mathbf{V}_1, \gamma_1)$ and $\Gamma_2 = (\mathbf{V}_2, \gamma_2)$ be CFA-views of $\mathbf{D}$, and let $f : \mathbf{V}_1 \to \mathbf{V}_2$ be any function which renders the diagram of Fig. 1 commutative. Then f is necessarily an open CFA-surjection, and hence defines a relative CFA-view $\Lambda(\Gamma_1, \Gamma_2) = (\mathbf{V}_2, \lambda\langle\Gamma_1, \Gamma_2\rangle)$ with $\lambda\langle\Gamma_1, \Gamma_2\rangle = f$.*

Proof. It is immediate that f is surjective, since γ_2 is. To show that it is open, let $M, N \in \mathsf{LDB}(\mathbf{V}_2)$ with $M \subseteq N$. Then, since γ_2 is open, there are $M', N' \in \mathsf{LDB}(\mathbf{D})$ with $M' \subseteq N'$ and $f(M') = M$, $f(N') = N$. Then $\gamma_1(M') \subseteq \gamma_1(N')$ with $\gamma_1(M') \in f^{-1}(M)$, $\gamma_1(N') \in f^{-1}(N)$; i.e., f is open. Finally, to show that it is basis preserving, let $N \in \mathsf{LDB}(\mathbf{V}_1)$; then, since γ_1 is surjective, there exists $M \in \mathsf{LDB}(\mathbf{D})$ with $\gamma_1(M) = N$. Since γ_1 is basis preserving, each $a \in N$ is of the form $\gamma_1(b)$ for some $b \in M$. Thus $f(N) = f(\gamma_1(M)) = \gamma_2(M) = \{\gamma_2(b) \mid b \in M\} = \{f(\gamma_1(b)) \mid b \in M\} = \{f(a) \mid a \in N\}$. □

Proposition 4.2. *Let $\mathbf{D} = (\mathsf{LDB}(\mathbf{D}), \subseteq)$ be a CFA-schema and let $\Gamma_1 = (\mathbf{V}_1, \gamma_1)$ and $\Gamma_2 = (\mathbf{V}_2, \gamma_2)$ be CFA-views of $\mathbf{D}$. Then there is a view morphism $f : \Gamma_1 \to \Gamma_2$ iff $\mathsf{Congr}(\Gamma_1) \subseteq \mathsf{Congr}(\Gamma_2)$.* □

In the order-based context, the congruences which define order views are the *order-compatible congruences* [2, 2.9] [3, 2.9]. In the present framework, the appropriate condition is that of being *atomically generated*; that is, of being

defined entirely by the behavior on the basis of the underlying schema. The formal details are as follows.

Definition 4.3 (Atomically generated equivalence relations). Let $\mathbf{D} = (\mathsf{LDB}(\mathbf{D}), \subseteq)$ be a CFA-schema, and let R be an equivalence relation on $\mathsf{LDB}(\mathbf{D})$. Define $[\mathbf{D}]_R = (\mathsf{LDB}(\mathbf{D})/R, \leqslant_{[\mathbf{D}]_\mathbf{R}})$, with $\mathsf{LDB}(\mathbf{D})/R$ the set of equivalence classes of R, and $\leqslant_{[\mathbf{D}]_\mathbf{R}}$ the relation on $\mathsf{LDB}(\mathbf{D})/R$ given by $[M]_R \leqslant_{[\mathbf{D}]_\mathbf{R}} [N]_R$ iff $(\exists M_1 \in [M]_R)(\exists N_1 \in [N]_R)(M_1 \leqslant_\mathbf{D} N_1)$, and define $\theta_R : \mathbf{D} \to \mathsf{LDB}(\mathbf{D})/R$ by $M \mapsto [M]_R$. In the order-based setting, it is already known that $[\mathbf{D}]_R$ is an order schema, and that $\Theta_R = ([\mathbf{D}]_R, \theta_R)$ is an order-based view [2, 2.9] [3, 2.9].

To extend this result to the CFA-setting, a few additional conditions must be imposed. The idea is that since CFA-morphisms are completely characterized by their action on atoms, the corresponding notion of equivalence relation should have this same property. Specifically, let $\mathbf{D} = (\mathsf{LDB}(\mathbf{D}), \subseteq)$ be a CFA-schema, and let R be an equivalence relation on $\mathsf{ExtAtoms}(\mathbf{D})$. Informally, R is atomically generated if its equivalence classes are defined entirely by the equivalences of its atoms. Formally, R is *atomically generated* if, for any $(M_1, M_2) \in \mathsf{LDB}(\mathbf{D}) \times \mathsf{LDB}(\mathbf{D})$, $(M_1, M_2) \in R$ iff the following two conditions are satisfied.

(atg:1) $(\forall A_1 \in \mathsf{Basis}_\mathbf{D}(M_1))(\exists A_2 \in \mathsf{ExtBasis}_\mathbf{D}(M_2))((A_1, A_2) \in R))$

(atg:2) $(\forall A_2 \in \mathsf{Basis}_\mathbf{D}(M_2))(\exists A_1 \in \mathsf{ExtBasis}_\mathbf{D}(M_1))((A_1, A_2) \in R))$

Put another way, define the *atomic subequivalence* $\mathsf{AtomicEq}(R) = R \cap (\mathsf{ExtAtoms}(\mathbf{D}) \times \mathsf{ExtAtoms}(\mathbf{D}))$. Then, for R to be atomically generated, it must be entirely recoverable from $\mathsf{AtomicEq}(R)$.

It is easy to see that an atomically generated equivalence relation provides the correct construction for obtaining an abstract FA-view; however, such a view is not concrete because the equivalence class $[M]_R$ is not the union of its basis. Nonetheless, this is easy to fix. For any $M \in \mathsf{LDB}(\mathbf{D})$, let $[\![M]\!]_R = \{[x]_R \mid x \in M\}$, let $\mathsf{LDB}(\mathbf{D})/\!\!/R = \{[\![M]\!]_R \mid M \in \mathsf{LDB}(\mathbf{D})\}$, and let $[\![\mathbf{D}]\!]_R = (\mathsf{LDB}(\mathbf{D})/\!\!/R, \subseteq)$. Define $[\![\Theta_R]\!] = ([\![\mathbf{D}]\!]_R, [\![\theta_R]\!])$, with $[\![\theta_R]\!] : \mathbf{D} \to [\![\mathbf{D}]\!]_R$ given by $M \mapsto [\![M]\!]_R$. This construction provides a CFA-view which is defined by the equivalence relation R; Lemma 4.4 and Proposition 4.5 below formalize this fact.

Lemma 4.4 (Concretization of views defined by equivalence relations). *Let $\mathbf{D}$ be a CFA-schema, and let R be an atomically generated equivalence relation on $\mathsf{LDB}(\mathbf{D})$. Then $[\![\mathbf{D}]\!]_R$ is a CFA-schema with $\mathsf{Atoms}([\![\mathbf{D}]\!]_\mathbf{R}) = \{[\![a]\!]_R \mid a \in \mathsf{Atoms}(\mathbf{D})\}$, and $[\![\Theta_R]\!] = ([\![\mathbf{D}]\!]_R, [\![\theta_R]\!])$ is a CFA-view of $\mathbf{D}$ with $\mathsf{Congr}([\![\Theta_R]\!]) = R$.* □

Proposition 4.5 (Characterization of CFA-views). *Let $\mathbf{D}$ be a CFA-schema, and let R be any equivalence relation on $\mathsf{LDB}(\mathbf{D})$. Then $[\![\Theta_R]\!]$ is a CFA-view iff R is an atomically generated equivalence. In particular, if $\Gamma = (\mathbf{V}, \gamma)$ is a CFA-view, then $\mathsf{Congr}(\Gamma)$ is an atomically generated equivalence.* □

A critical component of the theory is the ability to "lift" the constructions on a constrained schema $\mathbf{D}$ to the associated unconstrained schema $\overline{\mathbf{D}}$. This includes also morphisms between such schemata, views, and even equivalence relations induced by views. To begin, the notion of lifting a morphism is introduced.

Definition 4.6 (Extensions of CFA-morphisms to unconstrained databases). Given CFA-schemata $\mathbf{D}_1 = (\mathsf{LDB}(\mathbf{D}_1), \subseteq)$ and $\overline{\mathbf{D}}_2 = (\mathsf{LDB}(\mathbf{D}_2), \subseteq)$ and a CFA-morphism $f : \mathbf{D}_1 \to \mathbf{D}_2$, there is a unique natural extension to $\overline{f} : \overline{\mathbf{D}}_1 \to \overline{\mathbf{D}}_2$, which renders the diagram of Fig. 4 to the right commutative. Specifically, for any $M \in \mathsf{LDB}(\mathbf{D}_1)$, define $\overline{f}(M) = \bigcup\{f(\{x\}) \mid x \in M\}$. The following is easy to verify.

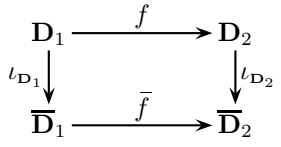

Fig. 4. The extension of a CFA-morphism

Proposition 4.7. *Let* $\mathbf{D}_1 = (\mathsf{LDB}(\mathbf{D}_1), \subseteq)$ *and* $\mathbf{D}_2 = (\mathsf{LDB}(\mathbf{D}_2), \subseteq)$ *be CFA-schemata, and let* $f : \mathbf{D}_1 \to \mathbf{D}_2$ *be a CFA-morphism. Then* $\overline{f} : \overline{\mathbf{D}}_1 \to \overline{\mathbf{D}}_2$ *is also a CFA-morphism, and it is a CFA-surjection iff* f *is.* □

Discussion 4.8 (Lifting of entire diagrams and completions of CFA-equivalences). The lifting construction described in Definition 4.6 may be applied to any commutative diagram containing CFA-morphisms. Thus, the commutative diagram shown in Fig. 5 to the right may be obtained from the commutative diagram shown in Fig. 3. The key results of this section, however, require a further step; namely, that the extension operation be moved from an entire construction to the individual components.

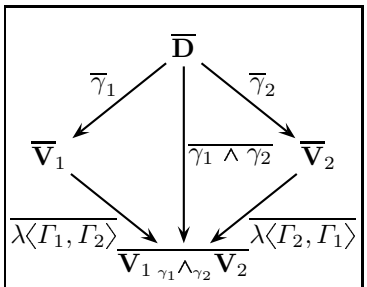

Fig. 5. Diagram lifting

In concrete terms, it is necessary to show that the diagram of Fig. 5 is the same, component-by-component and morphism-by-morphism, as that of Fig. 6 to the right. The key to establishing this equivalence lies within the associated equivalence relations. Specifically, given a CFA-schema $\mathbf{D}$, it is the case that the equivalence relation of $\overline{\mathbf{D}}$ is a natural completion of that of $\mathbf{D}$. More formally, proceed as follows. Let R be a CFA-equivalence on the CFA-schema $\mathbf{D}$. Define the *completion* of R to be the equivalence relation $\overline{R}$ on $\overline{\mathbf{D}}$ with the property that for $M_1, M_2 \in \mathsf{DB}(\mathbf{D})$,

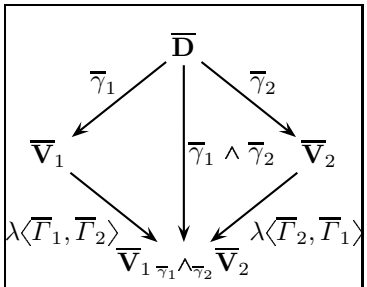

Fig. 6. Individual lifting

$$(M_1, M_2) \in \overline{R} \Leftrightarrow (((\forall A_1 \in \mathsf{Basis}_{\mathbf{D}}(M_1))(\exists A_2 \in \mathsf{ExtBasis}_{\mathbf{D}}(M_2))((A_1, A_2) \in R)) \wedge ((\forall A_2 \in \mathsf{Basis}_{\mathbf{D}}(M_2))(\exists A_1 \in \mathsf{ExtBasis}_{\mathbf{D}}(M_1))((A_1, A_2) \in R)))$$

Definition 4.9 (Independence dependencies). Consider a simple relational schema $R[ABC]$ constrained by the single FD $B \to C$, which decomposes losslessly and in a dependency-preserving fashion into the two projections Π_{AB} and

Π_{BC}. There are two equivalent ways of identifying the view states which may be combined to form a state of the main schema. First, one may say that the projection of each view on attribute C is the same. Second, on may say that for each tuple (a, b) of the view Π_{AB}, there is a tuple (b, c) with a matching B-value, and conversely. These conditions are so obviously identical that they are often considered as one. However, in a more general context, they display an important difference. The first (characterization by equivalent projections) does not make use explicit use of individual tuples, and thus claims a generalization as the *Π_C-independence dependency* within the framework of [2] [3]. On the other hand, a generalization of the second condition (tuple-by-tuple matching) requires a corresponding abstraction of the notion of a tuple; while the framework of [2] [3] does not admit such an abstraction, the more restricted one of used here does. The formalizations are as follows.

Let $\mathbf{D} = (\mathsf{LDB}(\mathbf{D}), \subseteq)$ be a CFA-schema, let $\{\Gamma_1, \Gamma_2\}$ be a subdirect complementary pair of CFA-views, and let $\Gamma_3 = (\mathbf{V}_3, \gamma_3)$ be a view of $\mathbf{D}$, with $\mathsf{Congr}(\Gamma_3) \subseteq \mathsf{Congr}(\Gamma_1)$ and $\mathsf{Congr}(\Gamma_3) \subseteq \mathsf{Congr}(\Gamma_2)$. The *$\Gamma_3$-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 $M_2 \in \mathsf{LDB}(\mathbf{V}_2)$, the following condition is satisfied ([2, 2.12], [3, 2.13]).

(id) $\quad ((M_1, M_2) \in \mathsf{LDB}(\mathbf{V}_1 \,{}_{\gamma_1}\!\otimes_{\gamma_2} \mathbf{V}_2)) \Leftrightarrow (\lambda\langle\Gamma_1, \Gamma_3\rangle(M_1) = \lambda\langle\Gamma_2, \Gamma_3\rangle(M_2))$

On the other hand, the *pointwise Γ_3-independence dependency* is satisfied iff the following two dual conditions are met.

(id:1)
$(\forall A_1 \in \mathsf{Basis}_{\mathbf{V}_1}(A_1))(\exists A_2 \in \mathsf{ExtBasis}_{\mathbf{V}_2}(A_2))(\lambda\langle\Gamma_1, \Gamma_3\rangle(A_1) = \lambda\langle\Gamma_2, \Gamma_3\rangle(A_2))$

(id:2)
$(\forall A_2 \in \mathsf{Basis}_{\mathbf{V}_1}(A_2))(\exists A_1 \in \mathsf{ExtBasis}_{\mathbf{V}_2}(A_1))(\lambda\langle\Gamma_1, \Gamma_3\rangle(A_1) = \lambda\langle\Gamma_2, \Gamma_3\rangle(A_2))$

Thus, in the context of CFA-views, conditions (id:1) and (id:2) may replace (id).

Proposition 4.10. *Let $(\mathsf{LDB}(\mathbf{D}), \subseteq)$ be a CFA-schema, and let $\{\Gamma_1, \Gamma_2\}$ be a subdirect complementary pair of CFA-views of $\mathbf{D}$. Then it is also a meet complementary pair iff conditions (id:1) and (id:2) of Definition 4.9 is satisfied.*

Proof. Follows directly from the discussion of Definition 4.9 and [2, 2.13] [3, 2.14]. □

Lemma 4.11 (Commuting congruences for completions). *Let $\mathbf{D}$ be a CFA-schema and let $M_1, M_2, \in \mathsf{DB}(\mathbf{D})$.*

(a) *Let $\Gamma = (\mathbf{V}, \gamma)$ be a CFA-view of $\mathbf{D}$. Then $(M_1, M_2) \in \mathsf{Congr}(\overline{\Gamma})$ iff the following two dual conditions are satisfied.*

$$(\forall A_1 \in \mathsf{Basis}_{\overline{\mathbf{D}}}(M_1))(\exists A_2 \in \mathsf{ExtBasis}_{\overline{\mathbf{D}}}(M_2))(A_1, A_2) \in \mathsf{Congr}(\Gamma))$$

$$(\forall A_2 \in \mathsf{Basis}_{\overline{\mathbf{D}}}(M_2))(\exists A_1 \in \mathsf{ExtBasis}_{\overline{\mathbf{D}}}(M_1))(A_1, A_2) \in \mathsf{Congr}(\Gamma))$$

(b) *Let $\Gamma_1 = (\mathbf{V}_1, \gamma_1)$ and $\Gamma_2 = (\mathbf{V}_2, \gamma_2)$ be FA-views of $\mathbf{D}$. Then $(M_1, M_2) \in \mathsf{Congr}(\overline{\Gamma}_1) \circ \mathsf{Congr}(\overline{\Gamma}_2)$ iff the following two dual conditions are satisfied.*

$$(\forall A_1 \in \mathsf{Basis}_{\overline{\mathbf{D}}}(M_1))(\exists A_2 \in \mathsf{ExtBasis}_{\overline{\mathbf{D}}}(M_2))(A_1, A_2) \in \mathsf{Congr}(\Gamma_1) \circ \mathsf{Congr}(\Gamma_2))$$

$$(\forall A_2 \in \mathsf{Basis}_{\overline{\mathbf{D}}}(M_2))(\exists A_1 \in \mathsf{ExtBasis}_{\overline{\mathbf{D}}}(M_1))(A_1, A_2) \in \mathsf{Congr}(\Gamma_1) \circ \mathsf{Congr}(\Gamma_2))$$

Proof. Part (a) is a direct consequence of conditions (atg:1) and (atg:2) of Definition 4.3 and the definition of completion. To establish (b), let $A_1 \in \mathsf{Basis}_{\overline{\mathbf{D}}}(M_1)$, $A_2 \in \mathsf{Basis}_{\overline{\mathbf{D}}}(M_2)$ with $(A_1, A_2) \in \mathsf{Congr}(\Gamma_1) \circ \mathsf{Congr}(\Gamma_2)$. Then, using the conditions of (b), there is an $N_{A_1A_2} \in \mathsf{DB}(\mathbf{D})$ with $(A_1, N_{A_1A_2}) \in \mathsf{Congr}(\Gamma_1)$ and $(N_{A_1A_2}, A_2) \in \mathsf{Congr}(\Gamma_2)$. $N_{A_1A_2}$ may furthermore be chosen to be a member of $\mathsf{ExtAtoms}(\mathbf{D})$, since every $A \in \mathsf{ExtBasis}_{\mathbf{D}}(N_{A_1A_2})$ must be equivalent to A_1 under $\mathsf{Congr}(\overline{\Gamma}_1)$ and equivalent to A_2 under $\mathsf{Congr}(\overline{\Gamma}_2)$, in view of (a). Thus, if $N_{A_1A_2} \neq \bot_{\mathbf{D}}$, any element of $\mathsf{Basis}_{\overline{\mathbf{D}}}(N_{A_1A_2})$ will serve as well as $N_{A_1A_2}$ itself. (If $N_{A_1A_2} = \bot_{\mathbf{D}}$, leave it as is.) Now, again by the characterization of (a), $(A_1, N_{A_1A_2}) \in \mathsf{Congr}(\overline{\Gamma}_1)$ and $(N_{A_1A_2}, A_2) \in \mathsf{Congr}(\overline{\Gamma}_2)$. Put $N' = \bigcup(\{N_{A_1A_2} \mid (A_1, A_2) \in \mathsf{Congr}(\overline{\Gamma}_1) \circ \mathsf{Congr}(\overline{\Gamma}_2)$ and $A_1 \in \mathsf{ExtBasis}_{\overline{\mathbf{D}}}(M_1)$ and $A_2 \in \mathsf{ExtBasis}_{\overline{\mathbf{D}}}(M_2)\}$. Then $(M_1, N') \in \mathsf{Congr}(\overline{\Gamma}_1)$ and $(N', M_2) \in \mathsf{Congr}(\overline{\Gamma}_2)$, whence $(M_1, M_2) \in \mathsf{Congr}(\overline{\Gamma}_1) \circ \mathsf{Congr}(\overline{\Gamma}_2)$. The converse condition follows immediately from part (a). □

Proposition 4.12 (Extension of commuting congruences). *Let* $\mathbf{D}$ *be a CFA-schema, and let* $\{\Gamma_1, \Gamma_2\}$ *be a fully commuting pair of CFA-views of* $\mathbf{D}$*. Then:*

(a) $\Gamma_1 \wedge \Gamma_2$ *is a CFA-view of* $\mathbf{D}$*.*
(b) $\{\overline{\Gamma}_1, \overline{\Gamma}_2\}$ *is a fully commuting pair of views of* $\overline{\mathbf{D}}$*, with* $\overline{\Gamma}_1 \wedge \overline{\Gamma}_2 = \overline{\Gamma_1 \wedge \Gamma_2}$*.*

Proof. The proof follows immediately from part (b) of Lemma 4.11. □

It is now possible to extend the notions of absolute k-models of Definition 3.6 to relative notions, and to prove the main theorem. A relative (to meet complement $\Gamma_2 = (\mathbf{V}_2, \gamma_2)$) k-model in the view $\Gamma_1 = (\mathbf{V}_1, \gamma_1)$ is a k-model of $\mathbf{V}_1$ whose $\Gamma_1 \wedge \Gamma_2$ component is already a legal database of $\mathbf{V}_{1\,\gamma_1 \wedge_{\gamma_2}} \mathbf{V}_2$. Such models are central to the update process because the property of the $\Gamma_1 \wedge \Gamma_2$ component being legal does not change under constant-complement update. The main theorem then states that, for the view to have this property, it suffices that the main schema have the k-submodel property.

Definition 4.13 (Relative k-models). *Let* $\mathbf{D} = (\mathsf{LDB}(\mathbf{D}), \subseteq)$ *be a CFA-schema, let* $\Gamma_1 = (\mathbf{V}_1, \gamma_1)$ *and* $\Gamma_2 = (\mathbf{V}_2, \gamma_2)$ *be CFA-views of* $\mathbf{D}$ *with* $\Gamma_2 \leqslant \Gamma_1$ *(i.e., with a morphism* $f : \Gamma_1 \to \Gamma_2$*) , and let* $k \in \mathbb{N}$*.*

(a) *The database* $M \in \mathsf{DB}(\mathbf{V}_1)$ *is called* Γ_2-legal *if* $\lambda\langle\overline{\Gamma}_1, \overline{\Gamma}_2\rangle(M) \in \mathsf{LDB}(\mathbf{V}_2)$*, and it is called a* Γ_2-relative k-model *for* $\mathbf{V}_2$ *if it is both* Γ_2*-legal and a* k*-model of* $\mathbf{V}_1$*.*
(b) *The view* $\Gamma_1 = (\mathbf{V}_1, \gamma_1)$ *has the* Γ_2-relative k-submodel property *if, for every* $M \in \mathsf{DB}(\mathbf{V}_1)$*,* $M \in \mathsf{LDB}(\mathbf{V}_1)$ *iff it is a* Γ_2*-relative* k*-model for* $\mathbf{V}_1$*.*

Theorem 4.14 (Preservation of complexity). *Let* $\mathbf{D}$ *be an FA-database schema, and let* $\{\Gamma_1 = (\mathbf{V}_1, \gamma_1), \Gamma_2 = (\mathbf{V}_2, \gamma_2)\}$ *be a subdirect complementary pair of CFA-views of* $\mathbf{D}$*. Let* $k \in \mathbb{N}$*. If* $\mathbf{D}$ *has the* k*-submodel property, then* Γ_1 *has the* $(\Gamma_1 \wedge \Gamma_2)$*-relative* k*-submodel property.*

Proof. First of all, note that γ_1, $\overline{\gamma}_1$, γ_2, and $\overline{\gamma}_2$ are downwardly strong and injectively generating, and that $\mathbf{V}_1$, $\widehat{\mathbf{V}}_1$, $\mathbf{V}_2$, and $\widehat{\mathbf{V}}_2$ are closed under subinstances, in view of Observation 3.7 and Proposition 3.11.

Let $M_1 \in \mathsf{DB}(\mathbf{V}_1)$ be a Γ_2-relative k-model; the goal is to establish that $M_1 \in \mathsf{LDB}(\mathbf{V}_1)$. To this end, begin by choosing $M_2 \in \mathsf{LDB}(\mathbf{V}_2)$ with the property that $\lambda\langle\overline{\Gamma}_1, \overline{\Gamma}_1 \wedge \overline{\Gamma}_2\rangle(M_1) = \lambda\langle\overline{\Gamma}_2, \overline{\Gamma}_1 \wedge \overline{\Gamma}_2\rangle(M_2) = \lambda\langle\Gamma_2, \Gamma_1 \wedge \Gamma_2\rangle(M_2)$. Such a choice of M_2 is possible since the view morphism $\lambda\langle\overline{\Gamma}_2, \overline{\Gamma}_1 \wedge \overline{\Gamma}_2\rangle$ is surjective. Observe that $(M_1, M_2) \in \overline{\mathbf{V}}_1 \underset{\overline{\gamma}_1}{\otimes}_{\overline{\gamma}_2} \overline{\mathbf{V}}_2$. Define $M = (\overline{\gamma}_1 \otimes \overline{\gamma}_2)^{-1}(M_1, M_2)$. Choose $N \in \mathsf{DB}(\mathbf{D})$ such that $N \subseteq M$ with two properties; first, that $\overline{\gamma}_1(N) = \overline{\gamma}_1(M)$, and, second, that $\overline{\gamma}_1$ is injective on $\mathsf{Basis}_{\overline{\mathbf{D}}}(N)$; i.e., $A_1, A_2 \in \mathsf{Basis}_{\overline{\mathbf{D}}}(N)$ and $\overline{\gamma}_1(A_1) = \overline{\gamma}_1(A_2)$ implies that $A_1 = A_2$. Such a choice is possible since $\overline{\gamma}_1$ is injectively generating. Now $(\overline{\gamma}_1 \otimes \overline{\gamma}_2)(N) = (M_1, M_2')$, with $M_2' \in \mathsf{LDB}(\mathbf{V}_2)$). ($M_2' \in \mathsf{LDB}(\mathbf{V}_2)$ since $M_2 \in \mathsf{LDB}(\mathbf{V}_2)$ and $\mathbf{V}_2$ is closed under subinstances.) Next, let $M'' \in \mathsf{DB}(\mathbf{D})$ with $M'' \subseteq N$ and $\mathsf{Card}(\mathsf{Basis}_{\mathbf{D}}(M'')) \leqslant k$. Define $(M_1'', M_2'') = (\overline{\gamma}_1 \otimes \overline{\gamma}_2)(M'')$. $M_2'' \in \mathsf{LDB}(\mathbf{V}_2)$ since $M_2'' \subseteq M_2'$ and $\mathbf{V}_2$ is closed under subinstances, and $M_1'' \in \mathsf{LDB}(\mathbf{V}_1)$ since $M_1'' \subseteq M_1$, $\mathsf{Card}(\mathsf{Basis}_{\mathbf{V}_1}(M_1'')) = \mathsf{Card}(\mathsf{Basis}_{\mathbf{D}}(M'')) \leqslant k$, and M_1 is a relative Γ_2-model. Thus, $M'' \in \mathsf{LDB}(\mathbf{D})$, since $\overline{\gamma}_1 \otimes \overline{\gamma}_2$ is an isomorphism. However, M'' was an arbitrary submodel of N in $\mathsf{DB}(\mathbf{D})$; thus, since $\mathbf{D}$ has the k-submodel property, $N \in \mathsf{LDB}(\mathbf{D})$. Finally, this implies that $M_1 = \gamma_1(N) \in \mathsf{LDB}(\mathbf{V}_1)$; whence Γ_1 has the $(\Gamma_1 \wedge \Gamma_2)$-relative k-model property. □

Example 4.15. It is instructive to give a detailed example at this point which illustrates the ideas of relative complexity. The example is $\mathbf{E}_2$, which was already introduced in Section 1. Specifically, let $\mathbf{E}_2$ denote the relational schema on five attributes with the single relation symbol $S[ABCDE]$. It is constrained by the set $\mathcal{F}_2 = \{A \rightarrow D, B \rightarrow D, CD \rightarrow A, A \rightarrow E\}$ of FDs. Since this schema is constrained by FDs, it clearly has the 2-submodel property. The view to be updated is $\Pi_{ABCE} = (S[ABCE], \pi_{ABCE})$, while the complement to be held constant is $\Pi_{ABCD} = (S[ABCD], \pi_{ABCD})$. The pair $\{\Pi_{ABCE}, \Pi_{ABCD}\}$ is lossless, since the dependency $A \rightarrow E$ implies the join dependency $ABCD \bowtie ABCE$, and it is dependency preserving since every FD in $\mathcal{F}_2$ embeds into one of the two views. Thus, it forms a meet-complementary pair, with meet $\Pi_{ABC} = (S[ABC], \pi_{ABC})$ [2, 2.16], [3, 2.17]. The updates which are allowed on Π_{ABCE} are precisely those which hold Π_{ABC} constant; that is, those which change only the E-value of a tuple. In view of the above theorem, Π_{ABCE} has the Π_{ABC}-relative 2-submodel property, since the main schema $\mathbf{E}_2$ has the 2-submodel property. Note that this is the case even though the view Π_{ABCE} cannot be finitely axiomatizable [4].

5 Update Strategies

To complete the transition to the CFA-context, the connection between the results of the previous section and formal update strategies must be made. For the most part, the approach is similar to that taken in [2] and [3]; however, an

adjustment is necessary to ensure that the complement view generated by an update strategy is a CFA-view.

Summary 5.1 (Augmenting update strategies for CFA-views). To adapt the conditions (upt:1)-(upt:8) summarized in Summary 2.1 and Summary 2.2 to the CFA-context, it is necessary to ensure that the equivalence $\equiv_\rho$ of an update strategy ρ is in fact an atomically generated equivalence, so that the ρ-complement $\tilde{\Gamma}^\rho$ of the CFA-view Γ is in fact a CFA-view. The appropriate addition to (upt:1)-(upt:8) is the following.

(upt:9) If $\rho(M_1, \gamma(M_2)) = M_2$, then

$$(\forall A_1 \in \mathsf{Basis}_{\mathbf{D}}(M_1))(\exists A_2 \in \mathsf{ExtBasis}_{\mathbf{D}}(M_2))(\rho(A_1, \gamma(A_2)) = A_2)$$

An update strategy ρ which satisfies all of (upt:1)-(upt:9) will be called a *CFA-update strategy*. Essentially, this means that every update is composed of updates on the underlying family of atoms. It is easy to see that this property holds in the classical setting of the lossless and dependency-preserving decomposition of a relational schema, as elaborated in [2, 2.15 and 2.16], [3, 2.16 and 2.17].

Lemma 5.2. *The induced update family $\equiv_\rho$ is an atomically generated equivalence iff ρ is a CFA-update strategy.*

Proof. Follows from Definition 4.3 and Proposition 4.5. □

Now the "CFA" equivalent of [2, 3.9] and [3, 3.10] follows directly.

Theorem 5.3. *Let $\mathbf{D}$ be a CFA-schema, and let Γ be a CFA-view of $\mathbf{D}$. There is natural bijective correspondence between CFA-update strategies for Γ and meet complements of that view which are also CFA-views. Specifically:*

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

(b) *For any meet complement Γ_1 of Γ which is also a CFA-view, $\tilde{\Gamma}^{\mathsf{UpdStr}\langle \Gamma, \Gamma_1 \rangle} = \Gamma_1$.* □

Notation 5.4 (Notational convention). Throughout the rest of this section, unless stated specifically to the contrary, let $\mathbf{D} = (\mathsf{LDB}(\mathbf{D}), \subseteq)$ be a CFA-schema, $\Gamma = (\mathbf{V}, \gamma)$ a CFA-view of $\mathbf{D}$, U and T closed update families for $\mathbf{D}$ and $\mathbf{V}$, respectively, and ρ a CFA-update strategy for T with respect to U.

Definition 5.5 (The completion of an update strategy).

(a) The *completion* of U, denoted $\overline{U}$, is the relation on $\mathsf{DB}(\mathbf{D}) \times \mathsf{DB}(\mathbf{D})$ defined by $(M_1, M_2) \in \overline{U}$ iff the following two (dual) conditions are satisfied.

(i) $(\forall A_1 \in \mathsf{Atoms}(\mathbf{M_1}))(\exists A_2 \in \mathsf{ExtAtoms}(\mathbf{M_2}))(A_1, A_2) \in U)$

(ii) $(\forall A_2 \in \mathsf{Atoms}(\mathbf{M_2}))(\exists A_1 \in \mathsf{ExtAtoms}(\mathbf{M_1}))(A_1, A_2) \in U)$

(b) The *completion* of ρ is the function $\overline{\rho} : \mathsf{DB}(\mathbf{D}) \times \mathsf{DB}(\mathbf{V}) \to \mathsf{DB}(\mathbf{D})$ given by $(M_1, N_2) \mapsto M_2$ iff $\overline{\gamma}(M_2) = N_2$ and the following two (dual) conditions are satisfied:

(i) $(\forall A_1 \in \mathsf{Atoms}(\mathbf{M_1}))(\exists A_2 \in \mathsf{ExtAtoms}(\mathbf{M_2}))(\rho(A_1, \gamma(A_2)) = A_2)$

(ii) $(\forall A_2 \in \mathsf{Atoms}(\mathbf{M_2}))(\exists A_1 \in \mathsf{ExtAtoms}(\mathbf{M_1}))(\rho(A_1, \gamma(A_2)) = A_2)$

Lemma 5.6. *$\overline{\rho}$ is a CFA-update strategy for $\overline{T}$ with respect to $\widehat{U}$.*

Proof. This is a routine verification against the conditions (upt:1)-(upt:9). The details are omitted. □

In [2, 4.2], [3, 4.3], it is established that there is only one way to reflect an update on a closed view back to the main schema, provided that update is realizable as sequence of legal insertions and deletions. Using the framework developed in this paper, it is possible to drop the condition of legality on the intermediate states; in other words, the reflection of the view update back to main schema is unique as long as it is realizable as sequence of insertions and deletions, even though the intermediate states may not be legal. In other words, for all practical purposes, there is only one way to reflect an update under a closed update strategy back to the main schema, regardless of whether or that update is order realizable. The formal details follow.

Definition 5.7 (Syntactic order-based updates). Following [2, 4.1] and [3, 4.1], a pair $(M_1, M_2) \in U$ is called a *formal insertion* with respect to U if $M_1 \leqslant_{\mathbf{D}} M_2$; a *formal deletion* with respect to U if $M_2 \leqslant_{\mathbf{D}} M_1$; and 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}), $1 \leqslant i \leqslant k-1$, is either a formal insertion or else a formal deletion with respect to U. The update family U is called *order realizable* if every pair in U is an order-based update.

More generally, call a pair $(M_1, M_2) \in U$ a *syntactic order-based update* if (M_1, M_2) is an order-based update in $\overline{U}$, and call U *syntactically order realizable* if every pair in U is a syntactic order-based update. Since $\overline{\rho}$ is an update strategy, the following extension of [2, 4.2] and [3, 4.3] follows immediately.

Theorem 5.8 (uniqueness of reflection of syntactic order-based view updates). *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 $(\gamma(M), N) \in T$ a syntactic order-based update, it must be the case that $\rho_1(M, N) = \rho_2(M, N)$. In particular, if T is syntactically order realizable, then $\rho_1 = \rho_2$.* □

Example 5.9. Let $\mathbf{E}_3$ be the relational schema with a single relation symbol $R[ABC]$ on three attributes, constrained by the set $\mathcal{F}_3 = \{B \to A, B \to C\}$. Let the view to be updated be $\Pi_{AB} = (R[AB], \pi_{AB})$, and the complement to be held constant $\Pi_{BC} = (R[BC], \pi_{BC})$. In view of [2, 2.16], [3, 2.17], these two views form a meet complementary pair with meet $\Pi_B = (R[B], \pi_B)$. The updates which are allowed on Π_{AB} are those which hold the projection on B constant; since the FD $B \to A$ holds as well, this means that the only updates which are possible are those which change the A-value of existing tuples. These are not order-based updates; therefore, the main theorem [2, 4.2] [3, 4.3] does not provide a direct guarantee of the uniqueness of their translations. However, when the integrity constraint $B \to A$ is ignored, the resulting update family is syntactically order based, and so Theorem 5.8 guarantees a unique translation

of all such updates on Π_{AB}, regardless of whether or not the complement to be held constant is Π_{BC}. Indeed, since the updates to Π_{AB} which hold Π_B constant are syntactically order realizable, the update strategy obtained by holding Π_{BC} constant is the only one possible.

This elegant solution should be contrasted with the rather complex and ad hoc approach to establishing uniqueness for the same example in [2, 4.5], [3, 4.6].

6 Final Remarks

Discussion 6.1 (Conclusions and proposed future work). It has been shown that, under quite general conditions, the explosion in constraint complexity which may occur when moving from a main schema to a view cannot adversely affect the complexity of updates issued against a closed database view. Essentially, such explosions in complexity must be encapsulated within the meet of the view to be updated and the complement used to define the update strategy. Since that part of the view is not alterable during an update, the complexity of the constraints on the meet are irrelevant. The complexity which is passed along to the view-update process is no greater than the corresponding complexity on the main schema.

The scope of the approach presented here is limited to a context which generalizes EGDs of the relational model, and covers neither TGDs such as join dependencies nor non-universal dependencies such as foreign-key constraints. In terms of practical use, the most salient task is to extend the framework to include foreign-key dependencies, since they are used in real, commercial relational database systems. To accomplish this, it seems necessary to extend the notion of a CFA-schema to one which explicitly recaptures the idea of a multi-relation schema, since such dependencies involve multiple relations in a fundamental way.

Extension to recapture TGDs is more straightforward, involving a generalization of the notion of k-model to (k_1, k_2)-model, with $k_1 \in \mathbb{N}$ and $k_2 \geqslant 1$ a real number. Roughly, $M \in \mathsf{DB}(\mathbf{D})$ is a (k_1, k_2)-model if there is a k_1-model $M' \in \mathsf{DB}(\mathbf{D})$ with $M \subseteq M'$ and $\mathsf{Card}(M') \leqslant k_2 \cdot \mathsf{Card}(M)$. Note that k-models are just $(k, 1)$-models in this extended context. Extension to recapture views defined by joins is also reasonably straightforward. While the view mappings are obviously no longer basis preserving, it is nonetheless possible to establish the necessary properties (i.e., those of Definition 3.10 and Proposition 3.11). All of these topics will be addressed in a forthcoming full version of this paper.

Finally, since the theory is not tied to any particular data model, it seems appropriate to apply this theory to models other than the classical relational. The difficulty is to find a suitable starting point, since the type of complexity questions addressed here have not been studied in any detail for models other than the relational.

Discussion 6.2 (Relationship to other work). In an early paper, Cosmadakis and Papadimitriou [14] present pessimistic complexity results which would appear to contradict those obtained here. However, they work with general

subdirect complements, and not meet complements, and so their results do not apply to the closed update strategies considered here. They also investigate the complexity of identifying a minimal (not necessarily meet) complement which will support a given update, again with pessimistic results. Recently, Lechtenbörger and Vossen [15] have also looked at the complexity of the problem of identifying (not necessarily meet) complements to views, but for the purpose of identifying information missing in the view, and not with an eye towards update strategies. Their approach, by design, does not concern itself with meet complements or update strategies. Beyond those works, most of the literature on the problem of complexity of view updates is focused on logic databases. The fundamental issues which arise in that context (theory-oriented database models) are quite different from those of instance-oriented database models, and so a meaningful comparison is difficult at best.

References

1. Bancilhon, F., Spyratos, N.: Update semantics of relational views. ACM Trans. Database Systems **6** (1981) 557–575
2. Hegner, S.J.: 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, Springer-Verlag (2002) 230–249
3. Hegner, S.J.: An order-based theory of updates for database views. Ann. Math. Art. Intell. **39** (2003) in press.
4. Hegner, S.J.: A simple counterexample to the finite axiomatizability of relational views. unpublished note, available at the author's web site (2003)
5. Adámek, J., Herrlich, H., Strecker, G.: Abstract and Concrete Categories. Wiley-Interscience (1990)
6. Langerak, R.: View updates in relational databases with an independent scheme. ACM Trans. Database Systems **15** (1990) 40–66
7. Hegner, S.J.: Foundations of canonical update support for closed database views. In Abiteboul, S., Kanellakis, P.C., eds.: ICDT'90, Third International Conference on Database Theory, Paris, France, December 1990, Springer-Verlag (1990) 422–436
8. Hegner, S.J.: Characterization of desirable properties of general database decompositions. Ann. Math. Art. Intell. **7** (1993) 129–195
9. Hegner, S.J.: Unique complements and decompositions of database schemata. J. Comput. System Sci. **48** (1994) 9–57
10. Davey, B.A., Priestly, H.A.: Introduction to Lattices and Order. second edn. Cambridge University Press (2002)
11. Grätzer, G.: General Lattice Theory. Second edn. Birkhäuser Verlag (1998)
12. Abiteboul, S., Hull, R., Vianu, V.: Foundations of Databases. Addison-Wesley (1995)
13. Wang, K., Graham, M.H.: Constant-time maintainability: A generalization of independence. ACM Trans. Database Systems **17** (1992) 201–246
14. Cosmadakis, S., Papadimitriou, C.: Updates of relational views. J. Assoc. Comp. Mach. **31** (1984) 742–760
15. Lechtenbörger, J., Vossen, G.: On the computation of relational view complements. ACM Trans. Database Systems **28** (2003) 175–208