Optimal Reflection of Bidirectional View Updates using Information-Based Distance Measures

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. When a database view is to be updated, there are generally many choices for the new state of the main schema. One way of characterizing the best such choice is to minimize the distance between the old state of the main schema and the new one. In recent work, a means of representing such distance based upon semantics was forwarded in which the distance between two states is measured by the the difference of *information* between the two, with the information of a state defined as the set of sentences from a particular set which hold on that state. This approach proved to be highly useful in identifying optimal reflections of insertions and to a lesser extent deletions, provided that the reflections were themselves insertions or deletions. In this work, that investigation is extended to *bidirectional* view updates – those which involve both insertion and deletion. It is shown that the definition of distance must be crafted more carefully in such situations. Upon so doing, a result is obtained which provides update reflections which are information optimal for insertion and deletion optimal with respect to tuples but not necessarily information.

1 Introduction

The problem of supporting updates to databases through views has long been one which has challenged researchers. Roughly speaking, the approaches may be considered to lie in one of three groups. The first, as represented by [7, 20, 21, 4], focus upon using the relational algebra, together with null values, to identify optimal or at least acceptable solutions. These approaches often handle certain cases very well but lack the thread of a unifying theory. The second group has focused upon the constant-complement approach, first forwarded by Bancilhon and Spyratos [3] and subsequently developed in [14, 15, 22]. The approach provides an elegant theory within a limited scope, but the conservative nature of constant-complement-based strategies leaves uncovered many important situations. The third approach, developed largely within the logic-programming community, is based upon using distance measures to find solutions which change the database as little as possible [2, 12, 13]. The core idea behind these approaches is to minimize the "distance" between the old state of the main schema and the new state which reflects the view update. Despite using logic as the tool for modelling, the distance is typically measured in a rather syntactic fashion, by minimizing the set of atoms, or even the number of atoms, which differ in the two databases. Such counting arguments fail to recapture that tuples are more than propositional atoms. For example, it is natural to expect the tuples $R(\mathrm{a}_1, \mathrm{b}_1, \mathrm{c}_1)$ and $R(\mathrm{a}_1, \mathrm{b}_1, \mathrm{c}_2)$ to be closer to each other than either is to $R(\mathrm{a}_3, \mathrm{b}_3, \mathrm{c}_3)$, but a measure which does not consider the inner structure of tuples cannot recapture this. To address this issue, more sophisticated measures of distance have been proposed [1] which are based upon (pseudo-)distance measures such as those proposed in [19] and [24]. These may then be extended to sets of tuples (i.e., databases) using aggregated measures, such as that of Eiter and Mannila [8]. Despite their obvious positive aspects, these measures nevertheless maintain a largely syntactic flavor.

In [17][1] and [16], an alternative approach is forwarded, in which a semantic notion of distance is employed, based upon the truth value of certain sentences rather than any syntactic characteristics of the atoms. The idea is to characterize a database state M by its information content $\mathsf{Info}\langle M, \Sigma\rangle$, the set of sentences in a set Σ which are true in M, and then to represent the distance between M and M' as the set of sentences in Σ whose truth values differ for the two states. The key issue is to choose Σ appropriately. It turns out that the most appropriate choice is $\mathsf{WFS}(\mathbf{D}, \exists\wedge+, K)$, the set of all existential, positive, and conjunctive sentences in the logic of the database schema $\mathbf{D}$ whose constant symbols are those of the set K consisting of all such symbols which occur in any of the current base state, the current view state, or the new view state. These sentences are just Boolean conjunctive queries [5]; i.e., conjunctive queries without free variables. A short example, adapted from that of [16, 1.3], will help illustrate the key ideas. For a more detailed presentation the reader is referred to that paper. Let $\mathbf{E}_0$ be the relational schema with relations $R[ABC]$ and $S[CD]$, constrained by the functional dependency (FD) $B \to C$ on $R[ABC]$ and the unary inclusion dependency (UIND) $R[C] \subseteq S[C]$, and let $M_{00} = \{R(\mathrm{a}_0, \mathrm{b}_0, \mathrm{c}_0), R(\mathrm{a}_1, \mathrm{b}_1, \mathrm{c}_1), S(\mathrm{c}_0, \mathrm{d}_0), S(\mathrm{c}_1, \mathrm{d}_1), S(\mathrm{c}_4, \mathrm{d}_4)\}$ be a database represented by ground atoms. Let $\Pi^{\mathbf{E}_0}_{R[AB]} = (R'[AB], \pi^{\mathbf{E}_0}_{R[AB]})$ be the view of $\mathbf{E}_0$ which projects $R[ABC]$ onto $R'[AB]$, dropping $S[A]$ entirely. Take M_{00} to be the initial state of schema $\mathbf{E}_0$; the corresponding view state is then $N_{00} = \{R'(\mathrm{a}_0, \mathrm{b}_0), R'(\mathrm{a}_1, \mathrm{b}_1)\}$. Now, suppose that the view update to insert $R'(\mathrm{a}_2, \mathrm{b}_2)$ is requested, so that the new view state will be $N_{01} = N_{00} \cup \{R'(\mathrm{a}_2, \mathrm{b}_2)\}$. The set of sentences over which information is measured is $\mathsf{WFS}(\mathbf{E}_0, \exists\wedge+, K_{00})$, with $K_{00} = \{\mathrm{a}_0, \mathrm{a}_1, \mathrm{a}_2, \mathrm{b}_0, \mathrm{b}_1, \mathrm{b}_2, \mathrm{c}_0, \mathrm{c}_1, \mathrm{d}_0, \mathrm{d}_1, \mathrm{c}_4, \mathrm{d}_4\}$ Any realization of this update must add $\varphi_{01} = (\exists x)(\exists y)(R(\mathrm{a}_2, \mathrm{b}_2, x) \wedge S(x, y))$ to the information content of M_{00}. A specific realization of this update must Skolemize the existentially quantified variables and add tuples such as those in $X_1 = \{(R(\mathrm{a}_2, \mathrm{b}_2, \bar{\mathrm{c}}_2), S(\bar{\mathrm{c}}_2, \bar{\mathrm{d}}_2)\}$ or $X_2 = \{(R(\mathrm{a}_2, \mathrm{b}_2, \bar{\mathrm{c}}_3), S(\bar{\mathrm{c}}_3, \bar{\mathrm{d}}_3)\}$, but since K_{00} does not contain any of the members of $\{\bar{\mathrm{c}}_2, \bar{\mathrm{d}}_2, \bar{\mathrm{c}}_3, \bar{\mathrm{d}}_3\}$, φ_{01} cannot distinguish between the solution which

[1] [17] contains some technical errors which have been corrected in the expanded version [16]. The reader is therefore referred to the latter paper.

adds X_1 from that which adds X_2. Thus, adding φ_{01} and its consequences in M_{00} to $\mathsf{Info}\langle M_{00}, \mathsf{WFS}(\mathbf{E}_0, \exists\wedge+, K_{00})\rangle$ characterizes the least added information necessary to realize the update, and so the distance from M_{00} to any Skolemized model of $M_{00} \cup \{\varphi_{01}\}$ must be least amongst all possibilities. Put another way, both $M_{00} \cup X_1$ and $M_{00} \cup X_2$ each have least distance from M_{00}; they differ only in the instantiation of variable by Skolem constants, and the logic cannot differentiate these instantiations. They are in a sense isomorphic solutions.

In [16], the focus is entirely upon unidirectional realizations of unidirectional update requests; that is, only view updates which are insertions and deletions are considered, and the reflection to the main schema must be of the same type as that of the request; insertions must reflected as insertions and deletions reflected as deletions. Suppose, now, that this restriction is dropped. The above solution is no longer optimal according to the definitions proposed. Indeed, the sentence $\varphi_{02} = \varphi_{01} \wedge S(\mathrm{c}_4, \mathrm{d}_4)$ is also added to $\mathsf{Info}\langle M_{00}, \mathsf{WFS}(\mathbf{E}_0, \exists\wedge+, K_{00})\rangle$ after the update, as are many others. This added information may be blocked by deleting $S(\mathrm{c}_4, \mathrm{d}_4)$, but that solution is not optimal either, since it deletes more information that the insert-only version. The addition of the sentence $S(\mathrm{c}_4, \mathrm{d}_4)$ to the inserted information is called a *collateral change*, because it is not part of the central update but rather a side effect of the way in which information change is measured. On the other hand, the sentence φ_{01} is essential to the update, and is called a *primary change*.

With an insertion which is to be reflected as an insertion, there is no problem with collateral changes. If φ_{01} is to be added, and $S(\mathrm{c}_4, \mathrm{d}_4)$ is already true, then it will still be true after the update. A similar observation applies to deletions. Of course, when the view update is an insertion, it is natural to require that it be reflected to the main schema as an insertion, and likewise for deletions. However, the problem of collateral change still remains for the class of view updates which involve both insertion and deletion.

The focus of this paper is to develop a formal framework for the representation of primary change which excludes collateral change, and to apply it to the characterization of the information-based distance between reflections, particularly in settings in which the update request involves both insertion and deletion. Notions of optimality based upon this measure are also developed.

2 The Underlying Context and Base Results

The underlying formalism for this paper is based heavily upon that which is developed in [16]. In this section, the essential terminology is reviewed, and a few useful extensions are presented.

Definition 2.1 (The relational model). All schemata are based upon a common *relational context* $\mathcal{D}$ which contains the attribute names and the constant names. Each domain has an infinite number of constants, but domains are allowed to overlap. Furthermore, there is a fixed *constant interpretation* $\mathcal{I}$ which enforces the unique name condition and which ensures that each domain value is bound to a constant. A *relational schema* $\mathbf{D}$ consists of a set of relation symbols, each with an arity, together with a set $\mathsf{Constr}(\mathbf{D})$ of integrity constraints. $\mathsf{WFF}(\mathbf{D})$ denotes the set of all well-formed formulas in the language of $\mathbf{D}$ with equality, while $\mathsf{WFF}(\mathbf{D}, \exists\wedge+, K)$ denotes the subset of $\mathsf{WFF}(\mathbf{D})$ consisting of those formulas which are existential (no universal quantifiers), positive (no negation), and conjunctive (no disjunction), and which involve only those constant symbols in the set K. If K consists of all constants, this will be shortened to $\mathsf{WFF}(\mathbf{D}, \exists\wedge+)$. $\mathsf{WFS}(\mathbf{D})$, $\mathsf{WFS}(\mathbf{D}, \exists\wedge+)$, and $\mathsf{WFS}(\mathbf{D}, \exists\wedge+, K)$ denote the corresponding sets of sentences (with no free variables). Integrity constraints are special sentences in $\mathsf{WFS}(\mathbf{D})$; see Definition 2.5 below.

Databases are modelled as *finite* sets of ground atoms. $\mathsf{DB}(\mathbf{D})$ denotes the set of all databases of $\mathbf{D}$ without regard to constraints, while $\mathsf{LDB}(\mathbf{D})$, the *legal databases* of $\mathbf{D}$, consists of those which satisfy $\mathsf{Constr}(\mathbf{D})$. For $\Phi \subseteq \mathsf{WFS}(\mathbf{D})$, $\mathsf{AtMod}_{\mathcal{I}}(\Phi)$ denotes the set of all "models" of φ — the set of databases M which satisfy φ in the sense that $M \cup \{\neg\varphi \mid \varphi \in \Phi\}$ is not satisfiable. For $\varphi \in \mathsf{WFS}(\mathbf{D})$, $\mathsf{AtMod}_{\mathcal{I}}(\varphi)$ is shorthand for $\mathsf{AtMod}_{\mathcal{I}}(\{\varphi\})$. $\models_{\mathbf{D}}$ denotes semantic entailment within $\mathbf{D}$; $\Phi \models_{\mathbf{D}} \varphi$ holds iff $\mathsf{AtMod}_{\mathcal{I}}(\Phi \cup \mathsf{Constr}(\mathbf{D})) \subseteq \mathsf{AtMod}_{\mathcal{I}}(\varphi)$.

A database mapping $f : \mathbf{D}_1 \to \mathbf{D}_2$ is represented as a logical interpretation; i.e., in the relational calculus. Thus, for each relation symbol R of $\mathbf{D}_2$, there is a formula f^R in the language of $\mathbf{D}_1$ with free variables corresponding exactly to the attributes of R. It is of *class* $\exists\wedge+$ if each of its defining formulas is in $\mathsf{WFF}(\mathbf{D}_2, \exists\wedge+)$. Projection, selection, join, and intersection are all of class $\exists\wedge+$, while union and difference are not. For t an atom, the *substitution* of t into f is the result of mapping t to a formula in $\mathsf{WFS}(\mathbf{D}_1)$. For example, if f is the view mapping $\pi^{\mathbf{E}_0}_{R[AB]}$ of the example of Section 1, and t is $R'(\mathrm{a}_1, \mathrm{b}_1)$, then $f^{R'} = (\exists x_C)(R(x_A, x_B, x_C))$, and $\mathsf{Subst}(f, t) = (\exists x_C)(R(\mathrm{a}_1, \mathrm{b}_1, x_C))$.

A *view* of the schema $\mathbf{D}$ is a pair $\Gamma = (\mathbf{V}, \gamma)$ in which $\mathbf{V}$ is the *view schema* and $\gamma : \mathbf{D} \to \mathbf{V}$ is a database mapping which is surjective on the underlying databases. The view Γ is of class $\exists\wedge+$ iff its mapping γ has this property.

Notation 2.2 ($\Upsilon^{\mathbf{D}}$ and $\Upsilon^{\mathbf{D}}_K$). Because they occur so frequently, especially as arguments in even larger formulas, the sets $\mathsf{WFS}(\mathbf{D}, \exists\wedge+)$ and $\mathsf{WFS}(\mathbf{D}, \exists\wedge+, K)$ will often be abbreviated to $\Upsilon^{\mathbf{D}}$ and $\Upsilon^{\mathbf{D}}_K$, respectively.

Definition 2.3 (Information content). Let $\mathbf{D}$ be a database schema, let $K \subseteq \mathsf{ConstSym}(D)$, and let $M \in \mathsf{DB}(\mathbf{D})$. The *information content* of M relative to $\Upsilon^{\mathbf{D}}_K$ is the set of all sentences in $\Upsilon^{\mathbf{D}}_K$ which are true for M. In other words, $\mathsf{Info}\langle M, \Upsilon^{\mathbf{D}}_K\rangle = \{\varphi \in \Upsilon^{\mathbf{D}}_K \mid M \in \mathsf{AtMod}_{\mathcal{I}}(\varphi)\}$. Each $\varphi \in \Upsilon^{\mathbf{D}}_K$ defines a Boolean conjunctive query on M; the information content consists of just those queries which are true. Note that information content is monotone; if $M_1 \subseteq M_2$, then $\mathsf{Info}\langle M_1, \Upsilon^{\mathbf{D}}_K\rangle \subseteq \mathsf{Info}\langle M_2, \Upsilon^{\mathbf{D}}_K\rangle$.

Definition 2.4 (Armstrong models over $\mathsf{WFS}(\mathbf{D}, \exists\wedge+, K)$). Let $\Phi \subseteq \Psi \subseteq \Upsilon^{\mathbf{D}}_K$ for some set finite K of constant symbols. The *closure* of Φ in Ψ, denoted $\mathsf{Closure}\langle \Phi, \Psi\rangle$, is $\{\varphi \in \Psi \mid \Phi \models \varphi\}$. An *Armstrong model* for Φ relative to Ψ is a model which satisfies the constraints of $\mathsf{Closure}\langle \Phi, \Psi\rangle$ but no other constraints of

Ψ. Armstrong models have been studied extensively for database dependencies; see, for example, [9] and [11]. The construction within $\Upsilon_K^{\mathbf{D}}$ is much simpler. For a finite set Φ, (almost) all that need be done is to rename variables so that all are distinct, remove the quantifiers, break the conjuncts into atoms, and then replace each variable with a distinct constant not occurring in K. For example, if $\Xi = \{R(\mathrm{a}_1, \mathrm{a}_2), R(\mathrm{a}_2, \mathrm{a}_3), (\exists x_1)(\exists x_2)(\exists x_3)(R(x_1, x_2) \wedge R(x_2, \mathrm{a}_3) \wedge R(\mathrm{a}_3, x_3))\}$. then $M_\Xi = \{R(\mathrm{a}_1, \mathrm{a}_2), R(\mathrm{a}_2, \mathrm{a}_3), R(\bar{\mathrm{a}}_1, \bar{\mathrm{a}}_2), R(\bar{\mathrm{a}}_2, \mathrm{a}_3), R(\mathrm{a}_3, \bar{\mathrm{a}}_3)\}$ is an Armstrong model of Ξ in $\Upsilon_K^{\mathbf{D}}$, with $K = \{\mathrm{a}_1, \mathrm{a}_2, \mathrm{a}_3\}$.

In this work, the need is for *canonical* Armstrong models which are reduced in the sense that they contain no redundancies. M_Ξ is not reduced; indeed, $M_\Xi = \{R(\mathrm{a}_1, \mathrm{a}_2), R(\mathrm{a}_2, \mathrm{a}_3), R(\mathrm{a}_3, \bar{\mathrm{a}}_3)\}$ is also an Armstrong model of Ξ which is canonical because it contains no redundancies. For details of the construction, including an elaboration of this example, see see [16, 3.3-3.10]. As a notational convenience, Skolem constants, that is, constants which replace variables in the construction of Armstrong models, will always be written with a bar above the name to distinguish them from constant symbols which appear in the source formulas.

Definition 2.5 (Generalized Horn dependencies and canonical models). In this work, the constraints of database schemata will always be *generalized Horn dependencies* (GHDs), as described in [16, 3.15]. They are very similar to, but a bit more general than, the database dependencies of [9]. They are of the form below, but are not required to be typed, may involve constant symbols, and allow trivial left-hand sides.

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

They include all traditional dependencies, such as functional dependencies (FDs) and inclusion dependencies (IDs). They are further partitioned into equality-generating (EGHDs) and tuple-generating (TGHDs). Every $\varphi \in \mathsf{WFS}(\mathbf{D}, \exists\wedge+)$ is a GHD in which the left-hand side of the rule is trivially true.

Let $\mathbf{D}$ be a database schema, K a finite subset of $\mathsf{Constr}(\mathbf{D})$, and $\Phi \subseteq \Upsilon_K^{\mathbf{D}}$. Define the *extended information* of Φ with respect to $\Upsilon_K^{\mathbf{D}}$ to be $\mathsf{XInfo}_{\mathbf{D}}\langle \Phi, \Upsilon_K^{\mathbf{D}} \rangle = \{\varphi \in \Upsilon_K^{\mathbf{D}} \mid \Phi \models_{\mathbf{D}} \varphi\}$ if $\Phi \cup \mathsf{Constr}(\mathbf{D})$ is consistent, it is precisely the set of sentences which every legal database which also satisfies Φ must satisfy. Typically, Φ will be of the form $M_1 \cup \Psi$, with M_1 the current database state and Ψ the set of sentences which are to be inserted to effect the update. Using an approach similar to the traditional chase procedure [23, 8.6-8.8], it can be shown that if $\mathsf{XInfo}_{\mathbf{D}}\langle \Phi, \Upsilon_K^{\mathbf{D}} \rangle$ is consistent, then it will always admit a model with least information provided the chase terminates [16, 3.20]. To ensure termination, it is sufficient that $\mathsf{Constr}(\mathbf{D})$ be *weakly acyclic* [10, Thm. 3.9]. Thus, under these conditions, an insertion to M_1 defined by a set $\Phi \subseteq \mathsf{WFS}(\mathbf{D}, \exists\wedge+, K)$ reflected from the view will always have a canonical least realization. Note that if Φ admits no model which is consistent with $\mathsf{Constr}(\mathbf{D})$, then $\mathsf{XInfo}_{\mathbf{D}}\langle \Phi, \Upsilon_K^{\mathbf{D}} \rangle = \Upsilon_K^{\mathbf{D}}$.

For $\Gamma = (\mathbf{V}, \gamma)$ a view of $\mathbf{D}$ of class $\exists\wedge+$ and $N \in \mathsf{DB}(\mathbf{D})$, $\mathsf{InfoLift}\langle N, \Gamma \rangle = \{\mathsf{Substf}\langle \gamma, t \rangle \mid t \in N\}$, the *lifting* of N to $\mathbf{D}$ along Γ. $\mathsf{XInfo}_{\mathbf{D}}\langle \mathsf{InfoLift}\langle N, \Gamma \rangle, \Upsilon^{\mathbf{D}} \rangle$ is the least information which must hold in every $M \in \mathsf{LDB}(\mathbf{D})$ with $N \subseteq \gamma(M)$.

3 Update Requests and Generators

Notation 3.1. For the rest of this paper, unless stated explicitly to the contrary, $\mathbf{D}$ will be taken to be a relational schema with $\mathsf{Constr}(\mathbf{D})$ a set of GHDs whose TGHDs are weakly acyclic, and $\Gamma = (\mathbf{V}, \gamma)$ will be taken to be a view of $\mathbf{D}$ which is of class $\exists\wedge+$. Recall also that $\Upsilon^{\mathbf{D}}$ and $\Upsilon^{\mathbf{D}}_K$ are shorthand for $\mathsf{WFS}(\mathbf{D}, \exists\wedge+)$ and $\mathsf{WFS}(\mathbf{D}, \exists\wedge+, K)$, respectively.

Definition 3.2 (Updates, update requests, and realizations). An *update* on $\mathbf{D}$ is a pair $\mu = (M_1, M_2) \in \mathsf{LDB}(\mathbf{D}) \times \mathsf{LDB}(\mathbf{D})$. M_1 is the current state, and M_2 the new state. If $M_1 \subseteq M_2$, μ is called an *insertion*, and, dually, if $M_2 \subseteq M_1$, μ is called a *deletion*. Collectively, insertions and deletions are termed *unidirectional updates*. An update which is not unidirectional; i.e., which includes both the insertion and the deletion of tuples, is called *bidirectional*.

Let (N_1, N_2) be an update on the schema $\mathbf{V}$ of Γ. A *reflection* (or *translation*) of (N_1, N_2) along Γ is an update (M_1, M_2) on $\mathbf{D}$ with $M_i = \gamma(N_i)$ for $i \in \{1, 2\}$. In the view update problem, the current state M_1 of the main schema is known, as is the view update (N_1, N_2); it is the new state M_2 of the main schema which is to be determined. Thus, as an economy of notation, define an *update request* from Γ to $\mathbf{D}$ as a pair $u = (M_1, N_2)$ in which $M_1 \in \mathsf{LDB}(\mathbf{D})$ (the old state of the main schema) and $N_2 \in \mathsf{LDB}(\mathbf{V})$ (the new state of the view schema). The pair u is called an *insertion request* (resp. a *deletion request*, resp. a *bidirectional request*) precisely in the case that (N_1, N_2) has that property. A *realization* of (M_1, N_2) along Γ is a reflection (M_1, M_2) of (N_1, N_2).

The set $\mathfrak{C}_u$ consists of all constant symbols which occur in any of M_1, N_1, N_2, $\mathsf{Constr}(\mathbf{D})$, and the formulas of the view mapping γ. The information content of the result M_2 of a realization will usually be measured in $\Upsilon^{\mathbf{D}}_{\mathfrak{C}_u}$. This accounts for all constant symbols in the source databases, as well as in the constraints, but ignores any new constants introduced by Skolemization in the construction of a canonical Armstrong model.

The view Γ *reflects insertions* (resp. *reflects deletions*) if every insertion request (resp. deletion request) has a realization which is also an insertion (resp. deletion). Γ is *strongly monotonic* if it reflects both insertions and deletions. See [16, Sec. 5] for a discussion of these concepts and conditions which guarantee that they hold.

Definition 3.3 (Full update difference). It is important to recall the definition of update difference which was forwarded in [16], and which works well for unidirectional updates but not for bidirectional ones. The *positive (Δ^+), negative (Δ^-), and total (Δ) full update differences* of $\mu = (M_1, M_2) \in \mathsf{LDB}(\mathbf{D}) \times \mathsf{LDB}(\mathbf{D})$ with respect to $\Upsilon^{\mathbf{D}}_K$ are defined as $\Delta^+\langle\mu, \Upsilon^{\mathbf{D}}_K\rangle = \mathsf{Info}\langle M_2, \Upsilon^{\mathbf{D}}_K\rangle \setminus \mathsf{Info}\langle M_1, \Upsilon^{\mathbf{D}}_K\rangle$, $\Delta^-\langle\mu, \Upsilon^{\mathbf{D}}_K\rangle = \mathsf{Info}\langle M_1, \Upsilon^{\mathbf{D}}_K\rangle \setminus \mathsf{Info}\langle M_2, \Upsilon^{\mathbf{D}}_K\rangle$, and $\Delta\langle\mu, \Upsilon^{\mathbf{D}}_K\rangle = \Delta^+\langle\mu, \Upsilon^{\mathbf{D}}_K\rangle \cup \Delta^-\langle\mu, \Upsilon^{\mathbf{D}}_K\rangle$, respectively. Note that, given $\varphi \in \Delta\langle\mu, \Upsilon^{\mathbf{D}}_K\rangle$, it is always possible to determine whether $\varphi \in \Delta^+\langle\mu, \Upsilon^{\mathbf{D}}\rangle$ or $\varphi \in \Delta^-\langle\mu, \Upsilon^{\mathbf{D}}\rangle$ by checking whether or not $M_1 \in \mathsf{AtMod}_{\mathcal{I}}(\varphi)$.

As noted in the introduction, the problem with this definition for bidirectional updates is that the sets $\Delta^+\langle\mu, \Upsilon^{\mathbf{D}}\rangle$ and $\Delta^-\langle\mu, \Upsilon^{\mathbf{D}}\rangle$ contain compound information, so that by adding elements to one, elements may be removed to the other. In the case of unidirectional updates, this is not an issue, since one of these sets will be empty. However, for bidirectional updates, both may be nonempty and this interference renders the measure less than completely useful.

Definition 3.4 (The semilattice $\Upsilon^{\mathbf{D}}/{\equiv_{\mathbf{D}}}$). There is a natural order and a natural equivalence induced by $\mathsf{Constr}(\mathbf{D})$ on $\Upsilon^{\mathbf{D}}$. To illustrate via example, in the schema $\mathbf{E}_0$ of Section 1, the inclusion dependency $R[C] \subseteq S[C]$ guarantees that whenever $(\exists x)(\exists y)(R(x, y, \mathrm{c}_0)$ is true in an arbitrary $M \in \mathsf{LDB}(\mathbf{E}_0)$, so too is $(\exists z)(S(\mathrm{c}_0, z))$. This is written as $(\exists x)(\exists y)(R(x, y, \mathrm{c}_0) \sqsubseteq_{\mathbf{E}_0} (\exists z)(S(\mathrm{a}_0, z))$. Similarly, the sentences $(\exists x)(\exists y)(R(x, y, \mathrm{c}_0)$ and $(\exists x)(\exists y)(R(x, y, \mathrm{c}_0) \wedge (\exists z)(S(\mathrm{a}_0, z))$ have identical truth values on all members of $\mathsf{LDB}(\mathbf{E}_0)$; this is written as $(\exists x)(\exists y)(R(x, y, \mathrm{c}_0) \equiv_{\mathbf{E}_0} (\exists x)(\exists y)(R(x, y, \mathrm{c}_0) \wedge (\exists z)(S(\mathrm{a}_0, z))$.

Formally, for the schema $\mathbf{D}$, define the preorder $\sqsubseteq_{\mathbf{D}}$ on $\Upsilon^{\mathbf{D}}$ by $\varphi_1 \sqsubseteq_{\mathbf{D}} \varphi_2$ iff $\varphi_2 \models_{\mathbf{D}} \varphi_1$. In other words, $\varphi_1 \sqsubseteq_{\mathbf{D}} \varphi_2$ iff φ_2 is stronger than φ_1 on legal databases. Define the equivalence relation $\equiv_{\mathbf{D}}$ on $\Upsilon^{\mathbf{D}}$ by $\varphi_1 \equiv_{\mathbf{D}} \varphi_2$ iff $\varphi_1 \sqsubseteq_{\mathbf{D}} \varphi_2 \sqsubseteq_{\mathbf{D}} \varphi_1$. Thus, $\equiv_{\mathbf{D}}$ identifies sentences which have identical truth values on all $M \in \mathsf{LDB}(\mathbf{D})$. The equivalence class of φ under $\equiv_{\mathbf{D}}$ is denoted by $[\varphi]_{\equiv_{\mathbf{D}}}$ or just $[\varphi]$, and the set of all such equivalence classes is $\Upsilon^{\mathbf{D}}/{\equiv_{\mathbf{D}}}$. Upon grouping equivalent elements, a partial order is obtained. Specifically, define $[\varphi_1] \sqsubseteq_{\mathbf{D}} [\varphi_2]$ to hold iff $\varphi_1 \sqsubseteq_{\mathbf{D}} \varphi_2$. It is easy to see that the partial order $\sqsubseteq_{\mathbf{D}}$ on $\Upsilon^{\mathbf{D}}/{\equiv_{\mathbf{D}}}$ defines a join-semilattice structure [6, Exer. 7.6] on $\sqsubseteq_{\mathbf{D}}$ with the join operation $\sqcup_{\mathbf{D}}$ given by $[\varphi_1] \sqcup_{\mathbf{D}} [\varphi_2] = [\varphi_1 \wedge \varphi_2]$.

Definition 3.5 (Ideals of $\Upsilon^{\mathbf{D}}/{\equiv_{\mathbf{D}}}$). Sets of sentences which occur in this work, such as those of the form $\mathsf{Info}\langle M, \Upsilon^{\mathbf{D}}_K\rangle$, are closed under implication within the context of the schema $\mathbf{D}$. The algebraic notion of an ideal of $\Upsilon^{\mathbf{D}}/{\equiv_{\mathbf{D}}}$ provides a suitable form for the representation of such sets in a compact fashion. Specifically, an *ideal* [6, Exer. 7.6] of $\Upsilon^{\mathbf{D}}/{\equiv_{\mathbf{D}}}$ is a subset which is closed downwards and under finite joins. More precisely, J is an ideal if (i) whenever $[\varphi_1] \in J$ and $[\varphi_2] \sqsubseteq_{\mathbf{D}} [\varphi_1]$, then $[\varphi_2] \in J$; and, (ii) whenever $[\varphi_1], [\varphi_2] \in J$, then $[\varphi_1] \sqcup_{\mathbf{D}} [\varphi_2] \in J$. Thus, ideals of $\Upsilon^{\mathbf{D}}/{\equiv_{\mathbf{D}}}$ are closed under $\models_{\mathbf{D}}$ and conjunction. The set of all ideals of $\Upsilon^{\mathbf{D}}/{\equiv_{\mathbf{D}}}$ is denoted $\mathsf{Ideals}(\Upsilon^{\mathbf{D}}/{\equiv_{\mathbf{D}}})$.

Ideals may be described compactly by the elements which generate them. Specifically, for $\Phi \subseteq \Upsilon^{\mathbf{D}}$, $\mathsf{Ideal}_{\mathbf{D}}(\Phi)$ is the smallest ideal of $\Upsilon^{\mathbf{D}}/{\equiv_{\mathbf{D}}}$ containing $\{[\varphi] \mid \varphi \in \Phi\}$. Formally, it is the intersection of all such ideals. It is useful to have a notation for extracting the underlying sentences from an ideal. For $J \subseteq \Upsilon^{\mathbf{D}}/{\equiv_{\mathbf{D}}}$, define $\|J\| = \{\varphi \mid [\varphi] \in J\}$.

Definition 3.6 (Specification of information change via ideals). To avoid the problem of collateral information change described in Section 1, the approach is to characterize $(\Delta^{+}\langle\mu, \Upsilon^{\mathbf{D}}_{\mathfrak{C}_u}\rangle, \Delta^{-}\langle\mu, \Upsilon^{\mathbf{D}}_{\mathfrak{C}_u}\rangle)$ as a pair of ideals. Formally, an *information-change specification* over $\mathbf{D}$ is given by an ordered pair $\langle G_{+}, G_{\not-}\rangle \in \mathsf{Ideals}(\Upsilon^{\mathbf{D}}/{\equiv_{\mathbf{D}}}) \times \mathsf{Ideals}(\Upsilon^{\mathbf{D}}/{\equiv_{\mathbf{D}}})$. G_{+} is the generator for the added information, and $G_{\not-}$ is the generator for the information which is preserved.

To identify the information change associated with such a specification, let $M \in \mathsf{LDB}(\mathbf{D})$ and let $\langle G_{+}, G_{\not-}\rangle$ be an information-change specification. The *new-state information* for M induced by $\langle G_+, G_{\not\top}\rangle$ relative to K is

$$\mathsf{NewInfo}_{\mathbf{D}}\langle M, \langle G_+, G_{\not\top}\rangle, \Upsilon^{\mathbf{D}}_K\rangle = \mathsf{XInfo}_{\mathbf{D}}\langle \|G_+\| \cup \|G_{\not\top}\|, \Upsilon^{\mathbf{D}}_K\rangle$$

Thus, the new information is just the sentences of G_+ and $G_{\not\top}$, closed up under the constraints of $\mathbf{D}$. The set K of constants is determined from the update realization and not by $\langle G_+, G_{\not\top}\rangle$ alone.

With this definition, it is possible to identify precisely the update which is induced by $\langle G_+, G_{\not\top}\rangle$. Let $u = (M_1, N_2)$ be an update request from Γ to $\mathbf{D}$, and let $\mu = (M_1, M_2)$ be a realization of u. The pair $\langle G_+, G_{\not\top}\rangle$ *generates* the realization μ for u if the following three conditions are satisfied.

(i) $G_+ \subseteq \mathsf{Ideal}_{\mathbf{D}}(\Delta^+\langle \mu, \Upsilon^{\mathbf{D}}_{\mathfrak{C}_u}\rangle)$.

(ii) $G_{\not\top} = \mathsf{Ideal}_{\mathbf{D}}(\mathsf{Info}\langle M_1, \Upsilon^{\mathbf{D}}\rangle \setminus \Delta^-\langle \mu, \Upsilon^{\mathbf{D}}_{\mathfrak{C}_u}\rangle)$.

(iii) $\mathsf{NewInfo}_{\mathbf{D}}\langle M_1, \langle G_+, G_{\not\top}\rangle, \Upsilon^{\mathbf{D}}_{\mathfrak{C}_u}\rangle = \mathsf{Info}\langle M_2, \Upsilon^{\mathbf{D}}_{\mathfrak{C}_u}\rangle$

There is no need for a subset representation in (ii) since collateral changes are limited to insertions. (Disjunction in the representation of information would be necessary for collateral changes in the deleted information to occur.)

This representation is similar in some ways to the $(\mathsf{Insert}, \mathsf{Retract})$ formalism of [2]. However, there are key differences. First, in [2] the elements of the update representation are ground atoms, whereas in the formalism of this paper they are sentences in $\Upsilon^{\mathbf{D}}_{\mathfrak{C}_u}$. Second, note that the generation of the deletion/retraction is given in complementary form — the set of sentences which are to be preserved, rather than the set of those which are to be deleted, is given. This a matter of convenience; it is much easier to represent the preserved information as an ideal than the deleted information as a filter, the dual of an ideal. Furthermore, special forms of updates, such as deletion optimality for tuples as presented in Definition 3.7 below, are much more succinctly specified via preservation.

Definition 3.7 (Optimality). The definitions of optimality are based upon subsumption in $\mathsf{Ideals}(\Upsilon^{\mathbf{D}}/\!\!\models_{\mathbf{D}})$. Let $u = (M_1, N_2)$ be an update request from Γ to $\mathbf{D}$, and let $\langle G_+, G_{\not\top}\rangle \in \mathsf{Ideals}(\Upsilon^{\mathbf{D}}/\!\!\models_{\mathbf{D}}) \times \mathsf{Ideals}(\Upsilon^{\mathbf{D}}/\!\!\models_{\mathbf{D}})$ generate a realization for u. The ideal G_+ is *insertion optimal* for u if for any other $\langle G'_+, G'_{\not\top}\rangle$ which defines a realization for u, $G_+ \sqsubseteq_{\mathbf{D}} G'_+$. Dually, $G_{\not\top}$ is *deletion optimal* for u if for any other $\langle G'_+, G'_{\not\top}\rangle$ which defines a realization for u, $G'_{\not\top} \sqsubseteq_{\mathbf{D}} G_{\not\top}$. Note the reversal of the inclusion for deletion optimality; a larger ideal preserves more and thus is to be preferred. Putting these together, $\langle G_+, G_{\not\top}\rangle$ is *(fully) optimal* for u if both G_+ is insertion optimal and $G_{\not\top}$ is deletion optimal for u.

Unfortunately, many update requests do not admit fully optimal solutions. There is, however, a restricted case which is often realizable. Define the *tuple ideals* of $\Upsilon^{\mathbf{D}}/\!\!\models_{\mathbf{D}}$ to be $\mathsf{Ideals}(\mathsf{DB}(\mathbf{D})) = \{\mathsf{Ideal}_{\mathbf{D}}(M) \mid M \in \mathsf{DB}(\mathbf{D})\}$. Thus, the tuple ideals are those which are generated by ground atoms; no quantifiers are allowed in the formulas. Call $G_{\not\top}$ *deletion optimal for tuples* if $G_{\not\top} \in \mathsf{Ideals}(\mathsf{DB}(\mathbf{D}))$ and for any other $\langle G'_+, G'_{\not\top}\rangle$ which defines a realization for u and for which $G'_{\not\top} \in \mathsf{Ideals}(\mathsf{DB}(\mathbf{D}))$, $G'_{\not\top} \sqsubseteq_{\mathbf{D}} G_{\not\top}$.

Examples 3.8. A few simple examples will help clarify these ideas. Let $\langle \mathbf{E}_0, \Pi^{\mathbf{E}_1}_{R[AB]}\rangle$ be the schema and view introduced in Section 1, with the current state of the main schema $M_{00} = \{R(\mathrm{a}_0, \mathrm{b}_0, \mathrm{c}_0), R(\mathrm{a}_1, \mathrm{b}_1, \mathrm{c}_1), S(\mathrm{c}_0, \mathrm{d}_0), S(\mathrm{c}_1, \mathrm{d}_1), S(\mathrm{c}_4, \mathrm{d}_4)\}$ and the current state of the view $N_{00} = \{R'(\mathrm{a}_0, \mathrm{b}_0), R'(\mathrm{a}_1, \mathrm{b}_1)\}$. Suppose that the desired new state of the view is $N_{02} = \{R'(\mathrm{a}_0, \mathrm{b}_0), R'(\mathrm{a}_2, \mathrm{b}_2)\}$, so that the update request is $u_{00} = (M_{00}, N_{02})$

Consider the two generators $\langle G^{01}_{+}, G^{01}_{\not-}\rangle$ and $\langle G^{02}_{+}, G^{02}_{\not-}\rangle$ for u_{00}, with $G^{01}_{+} = \mathsf{Ideal}_{\mathbf{E}_0}(\{(\exists z)(R(\mathrm{a}_2, \mathrm{b}_2, z))\})$, $G^{02}_{+} = \mathsf{Ideal}_{\mathbf{E}_0}(\{(R(\mathrm{a}_2, \mathrm{b}_2, \mathrm{c}_1))\})$, $G^{01}_{\not-} = \mathsf{Ideal}_{\mathbf{E}_{02}}(M_{00} \setminus \{R(\mathrm{a}_1, \mathrm{b}_1, \mathrm{c}_1)\})$, and $G^{02}_{\not-} = \mathsf{Ideal}_{\mathbf{E}_0}(\mathsf{Info}\langle M_{00}, \varUpsilon^{\mathbf{E}_0}\rangle \setminus \mathsf{Info}\langle\{(\exists z)(R(\mathrm{a}_1, \mathrm{b}_1, z))\}, \varUpsilon^{\mathbf{E}_0}\rangle)$. The pair $\langle G^{01}_{+}, G^{01}_{\not-}\rangle$ generates the update of M_{00} to $M_{01} = \{R(\mathrm{a}_0, \mathrm{b}_0, \mathrm{c}_0), R(\mathrm{a}_2, \mathrm{b}_2, \bar{\mathrm{c}}_2), S(\mathrm{c}_0, \mathrm{d}_0), S(\mathrm{c}_1, \mathrm{d}_1), S(\bar{\mathrm{c}}_2, \bar{\mathrm{d}}_2), S(\mathrm{c}_4, \mathrm{d}_4)\}$, while $\langle G^{02}_{+}, G^{02}_{\not-}\rangle$ generates $M_{02} = \{R(\mathrm{a}_0, \mathrm{b}_0, \mathrm{c}_0), R(\mathrm{a}_2, \mathrm{b}_2, \mathrm{c}_1), S(\mathrm{c}_0, \mathrm{d}_0), S(\mathrm{c}_1, \mathrm{d}_1), S(\mathrm{c}_4, \mathrm{d}_4)\}$. Roughly, $\langle G^{01}_{+}, G^{01}_{\not-}\rangle$ corresponds to deleting $R'(\mathrm{a}_0, \mathrm{b}_0)$ and inserting $R'(\mathrm{a}_2, \mathrm{b}_2)$, while $\langle G^{02}_{+}, G^{02}_{\not-}\rangle$ corresponds to the replacements $\mathrm{a}_1 \mapsto \mathrm{a}_2$ and $\mathrm{b}_1 \mapsto \mathrm{b}_2$ in $R'(\mathrm{a}_1, \mathrm{b}_1)$. The pair $\langle G^{01}_{+}, G^{01}_{\not-}\rangle$ is insertion optimal; this will be proven more generally in Proposition 4.3, but in this case that fact is easily seen by inspection. It is also deletion optimal for tuples, as will be established more generally in Theorem 4.6. However, it is not deletion optimal in general, since $\langle G^{02}_{+}, G^{02}_{\not-}\rangle$ deletes less information. More specifically, the sentence $(\exists x)(\exists y)(R(x, y, \mathrm{c}_1) \wedge S(\mathrm{c}_1, \mathrm{c}_1))$ is preserved with $\langle G^{02}_{+}, G^{02}_{\not-}\rangle$ but not $\langle G^{01}_{+}, G^{01}_{\not-}\rangle$. On the other hand, more information is inserted with $\langle G^{02}_{+}, G^{02}_{\not-}\rangle$ as well, since $R(\mathrm{a}_2, \mathrm{b}_2, \mathrm{c}_1) \models_{\mathbf{E}_0} (\exists z)(R(\mathrm{a}_2, \mathrm{b}_2, z))$, but not conversely. It is easy to see by inspection that $\langle G^{02}_{+}, G^{02}_{\not-}\rangle$ is deletion optimal. Since $\langle G^{01}_{+}, G^{01}_{\not-}\rangle$ is insertion optimal, no solution which is both insertion and deletion optimal can exist. In Theorem 4.6, it will be shown that under conditions satisfied by $\langle \mathbf{E}_0, \varPi^{\mathbf{E}_0}_{R[AB]}\rangle$, generators such as $\langle G^{01}_{+}, G^{01}_{\not-}\rangle$ which are insertion optimal without restriction as well as deletion optimal for tuples always exist.

It is not always the case that deletion-optimal generators exist. For example, define $N_{03} = \{R'(\mathrm{a}_2, \mathrm{b}_2)\}$, with the update request $u_{03} = (M_{00}, N_{03})$. There is an insertion-optimal generator $\langle G^{03}_{+}, G^{03}_{\not-}\rangle$, given by $G^{03}_{+} = \mathsf{Ideal}_{\mathbf{E}_0}(\{(\exists z)(R(\mathrm{a}_2, \mathrm{b}_2, z))\})$ and $G^{03}_{\not-} = \mathsf{Ideal}_{\mathbf{E}_0}(M_{03} \setminus \{R(\mathrm{a}_0, \mathrm{b}_0, \mathrm{c}_0), R(\mathrm{a}_1, \mathrm{b}_1, \mathrm{c}_1)\})$ with the resulting state of the form $M_{03} = \{R(\mathrm{a}_2, \mathrm{b}_2, \bar{\mathrm{c}}_2), S(\mathrm{c}_0, \mathrm{d}_0), S(\mathrm{c}_1, \mathrm{d}_1), S(\bar{\mathrm{c}}_2, \bar{\mathrm{d}}_2), S(\mathrm{c}_4, \mathrm{d}_4)\}$. However, there is no deletion-optimal generator. Indeed, consider the two states $M_{03} = \{R(\mathrm{a}_2, \mathrm{b}_2, \mathrm{c}_0), S(\mathrm{c}_0, \mathrm{d}_0), S(\mathrm{c}_1, \mathrm{d}_1), S(\mathrm{c}_2, \mathrm{d}_2), S(\mathrm{c}_4, \mathrm{d}_4)\}$ and $M'_{03} = \{R(\mathrm{a}_2, \mathrm{b}_2, \mathrm{c}_1), S(\mathrm{c}_0, \mathrm{d}_0), S(\mathrm{c}_1, \mathrm{d}_1), S(\mathrm{c}_2, \mathrm{d}_2), S(\mathrm{c}_4, \mathrm{d}_4)\}$, which are generated by $\langle G^{04}_{+}, G^{04}_{\not-}\rangle$ and $\langle G^{04\prime}_{+}, G^{04\prime}_{\not-}\rangle$ respectively, with $G^{04}_{+} = \mathsf{Ideal}_{\mathbf{E}_0}(\{R(\mathrm{a}_2, \mathrm{b}_2, \mathrm{c}_0)\})$, $G^{04\prime}_{+} = \mathsf{Ideal}_{\mathbf{E}_0}(\{R(\mathrm{a}_2, \mathrm{b}_2, \mathrm{c}_1))\})$, $G^{04}_{\not-} = \mathsf{Ideal}_{\mathbf{E}_0}(\mathsf{Info}\langle M_{03}, \varUpsilon^{\mathbf{E}_0}\rangle \setminus \mathsf{Info}\langle\{(\exists z)(R(\mathrm{a}_0, \mathrm{b}_0, z)), R(\mathrm{a}_1, \mathrm{b}_1, \mathrm{c}_1)\}, \varUpsilon^{\mathbf{E}_0}\rangle)$ and $G^{04\prime}_{\not-} = \mathsf{Ideal}_{\mathbf{E}_0}(\mathsf{Info}\langle M_{03}, \varUpsilon^{\mathbf{E}_0}\rangle \setminus \mathsf{Info}\langle\{(\exists z)(R(\mathrm{a}_1, \mathrm{b}_1, z)), R(\mathrm{a}_0, \mathrm{b}_0, \mathrm{c}_0)\}, \varUpsilon^{\mathbf{E}_0}\rangle)$. There is no way to include both $(\exists x)(\exists y)(R(x, y, \mathrm{b}_0))$ and $(\exists x)(\exists y)(R(x, y, \mathrm{b}_1))$ in a solution without violating the FD $B \rightarrow C$, since y must be bound b_2 in every R-tuple of a solution.

Example 3.9 (Full optimality). Optimal solutions do exist in certain situations, and it is instructive to illustrate one of them. Let $\mathbf{E}_1$ have the single relational symbol $R[ABC]$, constrained by the join dependency $\bowtie [AB, BC]$.

Define the view $\Pi^{\mathbf{E}_1}_{R[AB]} = (R'[AB], \pi^{\mathbf{E}_1}_{R[AB]})$ to be that which projects $R[ABC]$ onto $R'[AB]$. Let $M_{10} = \{R(\mathrm{a}_0, \mathrm{b}_0, \mathrm{c}_0), R(\mathrm{a}_1, \mathrm{b}_1, \mathrm{c}_1)\} \in \mathsf{LDB}(\mathbf{E}_1)$; the corresponding view state is then $N_{10} = \{R'(\mathrm{a}_0, \mathrm{b}_0), R'(\mathrm{a}_1, \mathrm{b}_1)\}$. Consider the update request $u_{10} = (M_{10}, N_{11})$ with $N_{11} = \{R'(\mathrm{a}_0, \mathrm{b}_0), R'(\mathrm{a}_2, \mathrm{b}_1)\}$ and the solution (M_{10}, M_{11}) with $M_{11} = \{R(\mathrm{a}_0, \mathrm{b}_0, \mathrm{c}_0), R(\mathrm{a}_2, \mathrm{b}_1, \mathrm{c}_1)\}$. This solution is optimal; indeed, it is generated by $\langle G^{10}_{+}, G^{10}_{\not-}\rangle$ with $G^{10}_{+} = \{(\exists z)(R(\mathrm{a}_2, \mathrm{b}_1, z))\}$ and $G^{10}_{\not-} = \mathsf{Ideal}_{\mathbf{E}_0}(\mathsf{Info}\langle M_{10}, \Upsilon^{\mathbf{E}_1}\rangle \setminus \mathsf{Info}\langle\{(\exists z)(R(\mathrm{a}_1, \mathrm{b}_1, z))\}, \Upsilon^{\mathbf{E}_1}\rangle)$, and it is easy to see that every solution must insert $(\exists z)(R(\mathrm{a}_2, \mathrm{b}_1, z))$ and delete $(\exists z)(R(\mathrm{a}_1, \mathrm{b}_1, z))$. This is an example of constant-complement update [3], [16], addressed further in Discussion 4.7.

4 Optimal Realization of Bi-Directional Update Requests

Example 4.1 (Motivating example). For any schema $\mathbf{D}$ satisfying the conditions spelled out in Notation 3.1, and any update request $u = (M_1, N_2)$, a sure way to obtain an insertion-optimal solution is to forget entirely what is M_1 and just use a canonical Armstrong model of $\mathsf{InfoLift}\langle N_2, \Gamma\rangle$. This is best illustrated by example; consider again the pair $\langle \mathbf{E}_0, \Pi^{\mathbf{E}_1}_{R[AB]}\rangle$ and the view state $N_{02} = \{R'(\mathrm{a}_0, \mathrm{b}_0), R'(\mathrm{a}_2, \mathrm{b}_2)\}$ of Examples 3.8. $\mathsf{InfoLift}\langle N_{02}, \Pi^{\mathbf{E}_0}_{R'[AB]}\rangle = \{(\exists z)(R(\mathrm{a}_0, \mathrm{b}_0, z)), (\exists z)(R(\mathrm{a}_2, \mathrm{b}_2, z))\}$, whose canonical Armstrong models in $\mathbf{E}_0$ are of the form $M_{02} = \{R(\mathrm{a}_0, \mathrm{b}_0, \bar{\mathrm{c}}_0), R(\mathrm{a}_2, \mathrm{b}_2, \bar{\mathrm{c}}_2)\}$. To make sure that the constant symbols $\bar{\mathrm{c}}_0$ and $\bar{\mathrm{c}}_2$ are not in $\mathfrak{C}_u$, with $u = (M, N_{02})$, it is necessary to know what the constant symbols of M are, but the construction of $\mathsf{InfoLift}\langle N_{02}, \Pi^{\mathbf{E}_0}_{R'[AB]}\rangle$ itself does not depend upon M or its constant symbols.

The generating pair may be written as $G_{04} = \langle G^{04}_{+}, G^{04}_{\not-}\rangle = \langle \mathsf{Ideal}_{\mathbf{E}_0}(\{(\exists z)(R(\mathrm{a}_0, \mathrm{b}_0, z)), (\exists z)(R(\mathrm{a}_2, \mathrm{b}_2, z))\}), \emptyset\rangle$, but this need not be an optimal representation. For example, if the initial state is $M_{00} = \{R(\mathrm{a}_0, \mathrm{b}_0, \mathrm{c}_0), R(\mathrm{a}_1, \mathrm{b}_1, \mathrm{c}_1), S(\mathrm{c}_0, \mathrm{d}_0), S(\mathrm{c}_1, \mathrm{d}_1), S(\mathrm{c}_4, \mathrm{d}_4)\}$, then $G^{04}_{\not-}$ specifies the deletion of $(\exists z)(R(\mathrm{a}_0, \mathrm{b}_0, x))$ while G^{04}_{+} then mandates its reinsertion. The optimal generator $G'_{04} = \langle G^{04\prime}_{+}, G^{04\prime}_{\not-}\rangle = \langle \mathsf{Ideal}_{\mathbf{E}_0}(\{(\exists z)(R(\mathrm{a}_2, \mathrm{b}_2, z))\}), \mathsf{Ideal}_{\mathbf{E}_0}(\mathsf{XInfo}_{\mathbf{E}_0}\langle M_{00}, \Upsilon^{\mathbf{E}_0}\rangle \setminus \mathsf{XInfo}_{\mathbf{E}_0}\langle\{(\exists z)(R(\mathrm{a}_0, \mathrm{b}_0, z))\}, \Upsilon^{\mathbf{E}_0}\rangle)\rangle$ avoids this problem. Returning to the general situation of $u = (M_1, N_2)$ on Γ and a realization $G = \langle G_{+}, G_{\not-}\rangle$, the information in $\mathsf{Info}\langle M_1, \Upsilon^{\mathbf{D}}\rangle$ which is also in $\mathsf{InfoLift}\langle N_2, \Gamma\rangle$ should be specified in $G_{\not-}$, not in G_{+}. The formalization of these ideas follows.

Definition 4.2. Let $u = (M_1, N_2)$ be an update request from Γ to $\mathbf{D}$. Define the *least insertion-optimal realization* $G^{\langle u+\rangle} = \langle G^{u+}_{+}, G^{u+}_{\not-}\rangle$ of u as

$$G^{\langle u+\rangle}_{+} = \mathsf{Ideal}_{\mathbf{D}}(\mathsf{InfoLift}\langle N_2, \Gamma\rangle \setminus \mathsf{Info}\langle M_1, \Upsilon^{\mathbf{D}}\rangle)$$
$$G^{\langle u+\rangle}_{\not-} = \mathsf{Ideal}_{\mathbf{D}}(\mathsf{Info}\langle M_1, \Upsilon^{\mathbf{D}}\rangle \cap \mathsf{InfoLift}\langle N_2, \Gamma\rangle)$$

Proposition 4.3. *Let $u = (M_1, N_2)$ be an update request from Γ to $\mathbf{D}$. Assume further that Γ reflects deletions. Then $G^{\langle u+\rangle}$ generates an insertion-optimal realization $\mu = (M_1, M_2)$ for u, with M_2 any canonical Armstrong model for*

$\mathsf{InfoLift}\langle N_2, \Upsilon\rangle$. *The update* μ *has the further property that for any other realization* $\mu' = (M_1, M_2')$, $\mathsf{Info}\langle M_2, \Upsilon^{\mathbf{D}}\rangle \subseteq \mathsf{Info}\langle M_2', \Upsilon^{\mathbf{D}}\rangle$.

Proof. It is clear that the construction produces the least information which a state $M_2 \in \mathsf{LDB}(\mathbf{D})$ for which $\gamma(M_2) = N_2$ must possess. The only concern is that $\gamma(M_2)$ may contain phantom tuples; that is, tuples which involve constant symbols resulting from the process of Skolemizing existentially quantified variables in the conversion from the information set to a canonical Armstrong model. The condition that Γ reflect deletions ensures that such tuples are impossible. For a discussion of this phenomenon, with examples, and a proof of the result that reflection of deletions prevents phantom tuples, consult [16, 4.8-4.11]. □

Definition 4.4 (Unit-head pairs). Tuple generating dependencies with more than one atom on the left-hand side create problems for deletions. If a rule of the form $A_1 \wedge A_2 \Rightarrow B$ holds, and B is to be deleted, then there is a choice of whether to delete A_1 or A_2, and so no least deletion exists. To obtain useful optimization results in the context of deletions, a restricted form of schema-view pair called a unit-head pair [16, 6.10] is appropriate. Since the database mapping γ of the view Γ also introduces constraints, it is first necessary to construct the *combined schema* $\mathsf{CombSch}\langle\mathbf{D}, \Gamma\rangle$, in which the main schema is augmented with the relation symbols of the view. For $\mathbf{E}_0$ introduced in Section 1, the relation symbol $R'[AB]$, as well as the constraint $(\forall x)(\forall y)(R'(x, y) \Leftrightarrow (\exists z)(R(x, y, z)))$ are added to the main schema. This constraint decomposes into the TGHDs $(\forall x)(\forall y)(R'(x, y) \Rightarrow (\exists z)(R(x, y, z)))$ and $(\forall x)(\forall y)(\forall z)(R(x, y, z) \Rightarrow R'(x, y))$. Now, call the pair $\langle\mathbf{D}, \Gamma\rangle$ *unit head* if each of the TGHDs of $\mathsf{CombSch}\langle\mathbf{D}, \Gamma\rangle$ is *unit head*; i.e., has at most one atom on the left-hand side. EGHDs are allowed without restriction, as are elements of $\Upsilon^{\mathbf{D}}$ and so-called mutual-exclusion dependencies of the form $A_1 \wedge \ldots \wedge A_k \Rightarrow \bot$. The pair $\langle\mathbf{E}_0, \Pi^{\mathbf{E}_0}_{R[AB]}\rangle$ of Section 1 and Examples 3.8 is unit head, while the pair $\langle\mathbf{E}_1, \Pi^{\mathbf{E}_1}_{R[AB]}\rangle$ of Example 3.9 is not.

Definition 4.5 (Reflection of general-source insertions). The notions of reflecting insertions and reflecting deletions play a central rôle in the characterization of views which support the reflection of unidirectional update requests in an optimal fashion. For bidirectional updates, a stronger version of insertion reflection is necessary, in which the source need not be a legal database but rather only a database which may be extended to a legal one. Formally, a *general-source insertion specification* is a pair $u = (M_1, N_2) \in \mathsf{DB}(\mathbf{D}) \times \mathsf{LDB}(\mathbf{V})$ with the property that (i) $\gamma(M_1) \subseteq N_2$ and (ii) there is some $M_1' \in \mathsf{LDB}(\mathbf{D})$ with $M_1 \subseteq M_1'$. A *realization* of u is a pair $(M_1, M_2) \in \mathsf{DB}(\mathbf{D}) \times \mathsf{LDB}(\mathbf{D})$ with the property that $M_1 \subseteq M_2$ and $\gamma(M_2) = N_2$. Say that Γ *reflects general-source insertions* if every general-source insertion specification admits a realization.

It is not known (at least not to the author) whether this condition is strictly stronger than insertion reflection [16, 4.13], which requires in addition that $M_1 \in \mathsf{LDB}(\mathbf{D})$ in the above. However, it is a simple exercise to show that it is satisfied by schemata which are constrained by FDs and UINDs (unary inclusion dependencies) with Γ is both FD-complete and UIND-complete. For a more complete presentation, see [16, Sec. 5].

Theorem 4.6. *Let $\langle \mathbf{D}, \Gamma \rangle$ be a unit-head pair which reflects deletions and general-source insertions, and let $u = (M_1, N_2)$ be an update request from Γ to $\mathbf{D}$. Define $P = \{t \in M_1 \mid \gamma(\{t\}) \subseteq N_2\}$. Then $\langle G_+^{\langle u+ \rangle}, \mathsf{Ideal}_{\mathbf{D}}(P) \rangle$ is both insertion optimal without restriction and deletion optimal for tuples for u.*

Proof. First note that because $\langle \mathbf{D}, \Gamma \rangle$ is unit-head, every interpretation formula γ^R of the view morphism γ consists of a single atom. Thus, γ is defined entirely by its action upon tuples; i.e., $\gamma(M) = \{\gamma(\{t\}) \mid t \in M\}$. Therefore, P as defined above is clearly the largest subset of M with the property that $\gamma(P) \subseteq N_2$; no larger subset $P' \subseteq M$ can possibly have the property that $\langle G_+^{\langle u+ \rangle}, \mathsf{Ideal}_{\mathbf{D}}(P') \rangle$ generates a realization for u. On the other hand, since Γ reflects general-source deletions and $\gamma(P) \subseteq N_2$, there must be an $M_2' \in \mathsf{LDB}(\mathbf{D})$ such that $P \subseteq M_2'$ and $\gamma(M_2') = N_2$. Since $\mathsf{XInfo}_{\mathbf{D}}\langle P \cup \mathsf{InfoLift}\langle N_2, \Gamma \rangle, \Upsilon^{\mathbf{D}} \rangle$ is the least information which any $M \in \mathsf{LDB}(\mathbf{D})$ which both contains P and satisfies $\mathsf{InfoLift}\langle N_2, \Gamma \rangle$ must have, $\mathsf{XInfo}_{\mathbf{D}}\langle P \cup \mathsf{InfoLift}\langle N_2, \Gamma \rangle, \Upsilon^{\mathbf{D}} \rangle \subseteq \mathsf{Info}\langle M_2', \Upsilon^{\mathbf{D}}_{\mathfrak{C}_u} \rangle$. Thus, $\mathsf{XInfo}_{\mathbf{D}}\langle P \cup \mathsf{InfoLift}\langle N_2, \Gamma \rangle, \Upsilon^{\mathbf{D}} \rangle$ is consistent and so admits a canonical Armstrong model M_2. This model must furthermore have the property that $\gamma(M_2) = N_2$. Indeed, if $\gamma(M_2)$ were to contain tuples not in N_2, they could be deleted using the update specification (M_2, N_2) and the fact that Γ reflects deletions. See [16, 4.10] for details surrounding this argument. That $\langle G_+^{\langle u+ \rangle}, \mathsf{Ideal}_{\mathbf{D}}(P) \rangle$ is insertion optimal follows from Proposition 4.3. □

See Examples 3.8 for examples of the above construction.

Discussion 4.7 (The optimality of constant-complement solutions). In [18], constant-complement update strategies ([3], [14]) are investigated the context of information-based distance measures, with the main result [18, 4.23] establishing that all such strategies based upon so-called semantically bijective decompositions are optimal. While this result is valid, the proof is in error in that it fails to take into account collateral information changes and works implicitly with the assumption that minimization of the size of a generator suffices. Fortunately, using the framework of the current paper, this use of generators may easily be made explicit and the repair of the proof of [18, 4.23] is almost trivial. The correct proof will appear in a revised version. *Thus, broadly stated, constant-complement update strategies are fully optimal.*

Example 4.8 (The limitations of semantic distance measures). Let $\mathbf{E}_2$ be the schema with two unary relation symbols $R[A]$ and $S[A]$, with no constraints, and let $\Pi^{\mathbf{E}_2}_{R[A]} = (R[A], \pi^{\mathbf{E}_2}_{R[A]})$ be the view of $\mathbf{E}_2$ which retains $R[A]$ entirely while discarding $S[A]$ completely. It is clear that any view update should keep $S[A]$ fixed; this is a simple example of the constant-complement strategy. Now, let $\mathbf{E}_3$ be identical to $\mathbf{E}_2$ save that a new relation symbol $T[A]$ is introduced with the constraint $(\forall x)(R(x) \wedge S(x) \Leftrightarrow T(x))$. The view $\Pi^{\mathbf{E}_3}_{R[A]} = (R[A], \pi^{\mathbf{E}_3}_{R[A]})$ is the same as $\Pi^{\mathbf{E}_2}_{R[A]}$. The two schemata $\mathbf{E}_2$ and $\mathbf{E}_3$ are logically equivalent, since $T[A]$ is defined completely in terms of $R[A]$ and $S[A]$. The isomorphism connecting them is even of class $\exists\wedge+$. Yet it is not clear that the optimal update strategies should be the same. If the state of $\mathbf{E}_3$ is $M_{31} = \{R(\mathrm{a}_0), S(\mathrm{a}_1)\}$, and the desired new view state is $N_{32} = \{R(\mathrm{a}_0), R(\mathrm{a}_1)\}$, then the optimal strategy, as defined by the theory of this paper, is to insert $R(\mathrm{a}_1)$ and keep $S[A]$ constant, which thus triggers an insertion of $T(\mathrm{a}_1)$. However, by deleting $S(\mathrm{a}_1)$, this insertion into $T[A]$ could be avoided.

In light of this example, one might argue that perhaps G_+ of an update generator $\langle G_+, G_{\not=} \rangle$ should only insert things which are not covered by inference; that is, to define $\mathsf{NewInfo}_{\mathbf{D}}\langle M, \langle G_+, G_{\not=}\rangle, \Upsilon_K^{\mathbf{D}}\rangle = \mathsf{XInfo}_{\mathbf{D}}\langle \|G_+\|, \Upsilon_K^{\mathbf{D}}\rangle \cup \mathsf{XInfo}_{\mathbf{D}}\langle \|G_{\not=}\|, \Upsilon_K^{\mathbf{D}}\rangle$; then $\{R(\mathrm{a}_1), R(\mathrm{a}_2), S(\mathrm{a}_2), T(\mathrm{a}_2)\}$ would no longer be preferred to $\{R(\mathrm{a}_1), R(\mathrm{a}_2)\}$ as a solution. Unfortunately, this strategy breaks the optimality of a constant-complement update defined by a join, such as in $\mathbf{E}_1$ of Example 3.9. Additional research is necessary to identify ways to retain the semantic nature of the proposed distance measures while respecting the kind of syntactic constructions illustrated in $\mathbf{E}_3$.

5 Conclusions and Further Directions

An approach to characterizing the distance between database states when both insertion and deletion are involved has been presented. In contrast to syntax-based approaches, it attempts to quantify the difference in meaning and thus adds a dimension which other distance measures lack.

On the other hand, as illustrated in Example 4.8, a semantics-based approach can sometimes produce questionable results (as can a syntax-based approach, of course). Therefore, an important next step in this research is to investigate ways to integrate both syntactic- and semantic-based measures to retain the best aspects of each. Methods for computing the distance between two states, at least for restricted classes of schemata, are also essential if this type of approach is to achieve practical utility.

References

1. O. Arieli, M. Denecker, and M. Bruynooghe. Distance semantics for database repair. *Ann. Math. Artif. Intell.*, 50(3-4):389–415, 2007.
2. O. Arieli, M. Denecker, B. V. Nuffelen, and M. Bruynooghe. Computational methods for database repair by signed formulae. *Ann. Math. Artif. Intell.*, 46(1-2):4–37, 2006.
3. F. Bancilhon and N. Spyratos. Update semantics of relational views. *ACM Trans. Database Systems*, 6:557–575, 1981.
4. F. Bentayeb and D. Laurent. Inversion de l'algèbre relationnelle et mises à jour. Technical Report 97-9, Université d'Orléans, LIFO, 1997.
5. A. K. Chandra and P. M. Merlin. Optimal implementation of conjunctive queries in relational data bases. In *Proceedings of the Ninth Annual ACM Symposium on Theory of Computing, 2-4 May 1977, Boulder, Colorado*, pages 77–90, 1977.
6. B. A. Davey and H. A. Priestly. *Introduction to Lattices and Order.* Cambridge University Press, second edition, 2002.

7. U. Dayal and P. A. Bernstein. On the correct translation of update operations on relational views. *ACM Trans. Database Systems*, 8(3):381–416, 1982.
8. T. Eiter and H. Mannila. Distance measures for point sets and their computation. *Acta Inf.*, 34(2):109–133, 1997.
9. R. Fagin. Horn clauses and database dependencies. *J. Assoc. Comp. Mach.*, 29(4):952–985, 1982.
10. R. Fagin, P. G. Kolaitis, R. J. Miller, and L. Popa. Data exchange: Semantics and query answering. *Theoret. Comput. Sci.*, 336:89–124, 2005.
11. R. Fagin and M. Y. Vardi. Armstrong databases for functional and inclusion dependencies. *Info. Process. Lett.*, 16:13–19, 1983.
12. J. A. Fernández, J. Grant, and J. Minker. Model theoretic approach to view updates in deductive databases. *J. Automated Reasoning*, 17(2):171–197, 1996.
13. S. Greco, C. Sirangelo, I. Trubitsyna, and E. Zumpano. Preferred repairs for inconsistent databases. In *7th International Database Engineering and Applications Symposium (IDEAS 2003), 16-18 July 2003, Hong Kong, China*, pages 202–211. IEEE Computer Society, 2003.
14. S. J. Hegner. An order-based theory of updates for closed database views. *Ann. Math. Art. Intell.*, 40:63–125, 2004.
15. S. J. Hegner. The complexity of embedded axiomatization for a class of closed database views. *Ann. Math. Art. Intell.*, 46:38–97, 2006.
16. S. J. Hegner. Information-based distance measures and the canonical reflection of view updates. Technical Report 0805, Institut für Informatik, Christian-Albrechts-Universität zu Kiel, October 2008. Also available on the web site of the author.
17. S. J. Hegner. Information-optimal reflections of view updates on relational database schemata. In S. Hartmann and G. Kern-Isberner, editors, *Foundations of Information and Knowledge Systems: Fifth International Symposium, FoIKS 2008, Pisa, Italy, February 11-15, 2008, Proceedings*, volume 4932 of *Lecture Notes in Computer Science*, pages 112–131. Springer-Verlag, 2008.
18. S. J. Hegner. Semantic bijectivity and the uniqueness of constant-complement updates in the relatiional context. In K.-D. Schewe and B. Thalheim, editors, *International Workshop on Semantics in Data and Knowledge Bases, SDKB 2008, Nantes, France, March 29, 2008, Proceedings*, volume 4925 of *Lecture Notes in Computer Science*, pages 172–191. Springer-Verlag, 2008.
19. A. Hutchinson. Metrics on terms and clauses. In M. van Someren and G. Widmer, editors, *Machine Learning: ECML-97, 9th European Conference on Machine Learning, Prague, Czech Republic, April 23-25, 1997, Proceedings*, volume 1224 of *Lecture Notes in Computer Science*, pages 138–145, 1997.
20. A. M. Keller. *Updating Relational Databases through Views*. PhD thesis, Stanford University, 1985.
21. R. Langerak. View updates in relational databases with an independent scheme. *ACM Trans. Database Systems*, 15(1):40–66, 1990.
22. J. Lechtenbörger. The impact of the constant complement approach towards view updating. In *Proceedings of the Twenty-Second ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, San Diego, California, June 09-11, 2003*, pages 49–55, 2003.
23. D. Maier. *The Theory of Relational Databases*. Computer Science Press, 1983.
24. S.-H. Nienhuys-Cheng. Distance between Herbrand interpretations: A measure for approximations to a target concept. In *Inductive Logic Programming, 7th International Workshop, ILP-97, Prague, Czech Republic, September 17-20, 1997, Proceedings*, volume 1297 of *Lecture Notes in Computer Science*, pages 213–226. Springer, 1997.