Enhancing the Updatability of Projective Views

(Extended Abstract)

Paolo Guagliardo[1], Reinhard Pichler[2], and Emanuel Sallinger[2]

[1] KRDB Research Centre, Free University of Bozen-Bolzano
[2] Vienna University of Technology

1 Introduction

Updating a database by means of a set of views is a classical problem in database research, known as the *view update problem*. It consists in "pushing back" the changes introduced into view relations by an update to the underlying database relations over which the view relations are defined. In very recent years, the view update problem has received renewed interest and attention [7,5,9,8,4].

View updates can be consistently and univocally propagated under the condition that the mapping between database and view relations is *lossless*, that is, the views provide the same amount of information as the database itself. Clearly, this is quite a strong requirement, as in common practical scenarios user views are created with the aim of allowing access only to specific portions of the database and thus, being lossy by design, such views are not directly updatable. The solution to this problem consists in using additional information, transferred by a so-called *view complement*, that is missing from the original view but needed to propagate the updates. However, under what is known as the *constant complement principle* [3], this complementary information must be invariant during the update process, that is, it must not be modified – directly or indirectly – by the view update. Therefore, the amount of information transferred by the view complement should be kept as small as possible.

To the best of our knowledge, in the relational setting, the only case studied in the literature is the one where the initial view consists of a single projection and the complement to be constructed is also restricted to a single projection [6]. A natural extension of the above, not investigated so far, is the case in which we are given a view as a *set of projections* and we want to construct a complement that consists of a set of projections too. The goal of our work is to initiate the generalization of the study of view complements in this direction.

Contribution and Outline. To properly choose a "good" view complement, we need a measure of how much information a complement provides w.r.t. another, together with an appropriate notion of *minimality*. In [6], where views and complements consist of a single projection, a minimal complement is one with the least number of attributes. Clearly, this notion of minimality is no longer

The research of R. Pichler and E. Sallinger is supported by the Austrian Science Fund (FWF):P25207-N23.

suitable when dealing with sets of projections. In Section 3, we thus start with the function-based order introduced in [3], and we observe that it coincides with the recently introduced *information transfer order* [2] on functional mappings. We show that, in general, there always exists a unique (up to information transfer equivalence) complement for any given view. However, this complement may not be expressible in a concrete view definition language, as is the case when views are defined by (even single) projections.

In Section 4, we restrict our attention to a setting where functional dependencies (FDs) are given on the source data and views are defined as sets of projections. We study in depth the information transferred by such views. For views defined by a *single projection*, the information transfer order between two views is easily reduced to the subset-relation between the sets of attributes of the two projections. To generalize this result to *sets of projections*, we present an algorithm that, in general, allows us to augment the set of attributes of the involved projections by taking the interaction between them into account.

Finally, in Section 5, we investigate the complement of a set of projections. To this end, we show that it may be beneficial to split a complement consisting of a single projection, as in [6], into a set of smaller projections. We thus present an algorithm that refines a single-projection complement and yields a set of projections that, in general, transfers less (or the same, but never more) information than the initial complement.

2 Preliminaries

A *schema* is a finite set of relation symbols. Let **dom** be an arbitrary (possibly infinite) set of domain values. An *instance* I of a schema $\mathbf{S}$ maps each relation symbol S in $\mathbf{S}$ to a relation S^I on **dom** of appropriate arity, called the *extension* of S under I. The set of elements of **dom** that occur in an instance I is called the *active domain* of I and is denoted by $\mathbf{adom}(I)$. All instances in this paper are finite, that is, have a finite active domain.

A *view* from $\mathbf{R}$ to $\mathbf{V}$ is a function associating instances of a *database schema* $\mathbf{R}$ with instances of a *view schema* $\mathbf{V}$ disjoint with $\mathbf{R}$. A view is *specified* in a query language $\mathcal{L}$ when each view symbol $V \in \mathbf{V}$ is defined by an $\mathcal{L}$-query over $\mathbf{R}$. For a set of integrity constraints Σ on the database schema $\mathbf{R}$, we denote by $\mathbf{R}_\Sigma$ the set of instances of $\mathbf{R}$ satisfying Σ, and we refer to such instances as *legal*. Then, a view *under* Σ associates legal instances of $\mathbf{R}$ with instances of $\mathbf{V}$.

In what follows, unless otherwise specified, views are assumed to be from the same database schema to distinct view schemas disjoint with each other, e.g., f and g are from $\mathbf{R}$ to $\mathbf{V}$ and from $\mathbf{R}$ to $\mathbf{W}$, respectively, with $\mathbf{V}$ and $\mathbf{W}$ disjoint with each other (and with $\mathbf{R}$). The amount of information that a view transfers from the source database w.r.t. another view can be measured as follows:

Definition 1 ([3]). *Let f and g be views under Σ. Then, f is* less informative *than g (written $f \leq g$) if, for every $I, I' \in \mathbf{R}_\Sigma$, $g(I) = g(I')$ implies $f(I) = f(I')$.*

We refer to the preorder $\leq$ as the *information-transfer order*, and we denote by $\equiv$ the associated equivalence relation ($f \equiv g$ if and only if $f \leq g$ and $g \leq f$). Observe that $f \leq g$ if and only if there exists a function h such that $f = h \circ g$, thus $\leq$ coincides with the order $\preceq_s$ of [2] for mappings that are functional.

A view loses information when it associates distinct database instances with the same view instance. This loss of information can be recovered by a view that "complements" the original one by separating instances the latter does not.

Definition 2 (View complement [3]). *Let f and g be views under Σ. Then, g is a* complement *of f if, for every distinct $I, I' \in \mathbf{R}_\Sigma$, $g(I) \neq g(I')$ whenever $f(I) = f(I')$. Moreover, g is* minimal *if, for every complement h of f, $g \leq h$ whenever $h \leq g$.*

Note that the above notion is symmetric, in that if g is a complement of f, then f is a complement of g. For this reason, sometimes we simply say that two views f and g are *complementary*.

3 Minimal Complements

In this section we present some general results about the minimality of complements, without committing to any particular language for specifying views. First, we show that a minimal complement complements the original view *only when needed*, as formally stated below.

Proposition 1. *Let f be a view under Σ, and let g be a minimal complement of f. Then, for every distinct I, I', $g(I) \neq g(I')$ if and only if $f(I) = f(I')$.*

This characterisation allows us to prove that there always exists a unique (up to information-transfer equivalence) minimal complement of any given view.

Theorem 1 (Unique minimal complement). *Let f be a view, and let g and h be minimal complements of f. Then, $g \equiv h$.*

In general, the unique minimal complement might not be expressible in the concrete language used for specifying views. In this case, we are interested in complements which are minimal among those expressible within the language. For a class $\mathfrak{C}$ of views specified in a language $\mathcal{L}$, minimal complements within $\mathfrak{C}$ are referred to as $\mathfrak{C}$-*minimal*. A unique $\mathfrak{C}$-minimal complement might no longer exist, as we show below for *projective views*, that is, views specified by projections.

Proposition 2. *Let $\mathfrak{C}$ be the class of projective views. There exist constraints Σ and views f, g and h in $\mathfrak{C}$ under Σ such that g and h are $\mathfrak{C}$-minimal complements of f which are incomparable under $\leq$.*

Proof. Let R be a relation symbol on attributes A, B, C, and let Σ consist of the FDs $A \to C$ and $B \to C$. Let the view symbols V_1, V_2 and V_3 be defined by projections on AB, BC and AC, respectively, and for $i \in \{1, 2, 3\}$ let f_i be

the corresponding views under Σ from $\mathbf{R} = \{R\}$ to $\mathbf{V}_i = \{V_i\}$. Then, f_2 and f_3 are both complements of f_1 because R is equivalent to the natural join of V_1 with V_2 and of V_1 with V_3, due to the FDs in Σ. The projection on ABC is obviously not minimal since it includes the attributes of V_2 and V_3, and no other projection is a complement. Hence, f_2 and f_3 are minimal. Let $I = \{R(a, b, c)\}$, $I' = \{R(a', b, c)\}$ and $I'' = \{R(a, b', c)\}$. Since $f_2(I) = f_2(I')$ and $f_3(I) \neq f_3(I')$, we have that $f_3 \not\leq f_2$ and, as $f_3(I) = f_3(I'')$ and $f_2(I) \neq f_2(I'')$, we also have that $f_3 \not\leq f_2$. Therefore, f_2 and f_3 are incomparable under $\leq$.

4 Information Transferred by Projective Views

In the rest of the paper, we consider a database schema consisting of only one relation symbol R on a set U of attributes, view symbols defined by projections on subsets of U and database constraints Σ given by FDs. W.l.o.g. we assume the FDs to be of the form $X \to A$, with $X \subseteq U$ and $A \in U$. We denote by $f^{\Sigma}_{X_1,\ldots,X_n}$ the view under Σ specified by the n projections on the sets of attributes $X_1, \ldots, X_n$. For ease of notation, we omit the superscript whenever there is no need to explicitly mention the set of FDs under consideration.

First we observe that, when views are single projections, the order $\leq$ can be characterised in terms of inclusion between the sets of attributes in the projections.

Proposition 3. *$f_X \leq f_Y$ if and only if $X \subseteq Y$.*

To compare the information transferred by several views defined by sets of projections it suffices to consider a single projection on the left-hand side.

Proposition 4. *$f_{X_1,\ldots,X_n} \leq f_{Y_1,\ldots,Y_m}$ iff $f_{X_i} \leq f_{Y_1,\ldots,Y_m}$ for each $i \in \{1, \ldots, n\}$.*

Towards an effective criterion for checking $\leq$, let us examine $S = \bowtie_{i=1}^{m} \pi_{Y_i}(I)$ for some instance I. In particular, the join can be computed by (1) picking m tuples $t_1, \ldots, t_m$ from I, (2) projecting them on the respective Y_i, i.e., $\pi_{Y_1}(t_1), \ldots, \pi_{Y_m}(t_n)$ and (3) checking the join conditions. Let $k = |\cup_{i=1}^{m} Y_i|$, then

$$S = \{\, (s_1, \ldots, s_k) \mid \exists t_1, \ldots, t_m \in I \text{ s.t. } \forall Y_i, A_j \colon A_j \in Y_i \implies t_i[A_j] = s_j \,\}$$

The above characterisation of S in terms of the equalities between a result tuple $(s_1, \ldots, s_k)$ and the attributes of the t_i's motivates Algorithm 1, which has the following property.

Theorem 2. *Let $\{Y_1^*, \ldots, Y_m^*\} = \mathsf{Saturate}(\{Y_1, \ldots, Y_m\}, \Sigma)$. If for every $i \in \{1, \ldots, n\}$ there is a $j \in \{1, \ldots, m\}$ s.t. $X_i \subseteq Y_j^*$, then $f^{\Sigma}_{X_1,\ldots,X_n} \leq f^{\Sigma}_{Y_1,\ldots,Y_m}$.*

As a corollary of the preceding theorem, it follows that Algorithm 1 preserves the information-transfer order.

Corollary 1. *Let $\{Y_1^*, \ldots, Y_m^*\} = \mathsf{Saturate}(\{Y_1, \ldots, Y_m\}, \Sigma)$. Then it holds that $f^{\Sigma}_{Y_1^*,\ldots,Y_m^*} \leq f^{\Sigma}_{Y_1,\ldots,Y_m}$.*

Algorithm 1 Saturation

```
INPUT:  Sets of attributes Y_1, ..., Y_n and a set of FDs Σ.
OUTPUT: Saturated sets of attributes Y_1^*, ..., Y_n^*.
 1: procedure Saturate({Y_1, ..., Y_n}, Σ)
 2:     for all i ∈ {1, ..., n} and j ∈ {1, ..., r}, where r is the arity of R do
 3:         if A_j ∈ Y_i then
 4:             t_i[A_j] = a_j   // a constant
 5:         else
 6:             t_i[A_j] = x_{i,j}   // a fresh existential variable
 7:         end if
 8:     end for
 9:     Let {t_1^*, ..., t_n^*} = chase({t_1, ..., t_n}, Σ)
10:     for all i ∈ {1, ..., n} do
11:         Let Y_i^* = {A_j | t_i^*[A_j] = a_j}
12:     end for
13:     return {Y_1^*, ..., Y_n^*}
14: end procedure
```

Note that Theorem 2 corresponds to the soundness of Algorithm 1. Yet, we conjecture that the following holds:

Conjecture 1. $f_X \leq f_{Y_1,\ldots,Y_m}$ if and only if, for every legal instance I holds

$$\pi_X(I) \supseteq \pi_X\big(\pi_{Y_1}(I) \bowtie \ldots \bowtie \pi_{Y_m}(I)\big) \tag{1}$$

This would amount to the completeness of Algorithm 1. We have a proof of Conjecture 1 for the special case $m = 2$, i.e., the right-hand side is given by two projections. The general form of Conjecture 1 as given above is an open question for future work.

5 Complements of Projective Views

Checking whether a view $f_{X_1,\ldots,X_n}$ is lossless under a set Σ of full dependencies (that is, EGDs and full TGDs) amounts to checking whether Σ entails the join dependency $\bowtie[X_1, \ldots, X_n]$ (we use the notation of [1]). As this can be done in polynomial time [1], we get the following.

Proposition 5. *When view complements are restricted to a single projection, a minimal complement of a view $f_{X_1,\ldots X_n}$ under full dependencies can be found in polynomial time.*

Proof. The algorithm is the same as for finding a minimal one-projection complement of a one-projection view [6]: Start with the trivial complement f_U and, examining the attributes in some arbitrary order, repeatedly remove any one of them that can be removed without affecting complementarity.

Algorithm 2 Refine complement

`INPUT:` Sets of attributes $X_1, \ldots, X_n$ and Y, and a set of FDs Σ.
`OUTPUT:` Refined sets of attributes $Y_1, \ldots, Y_m$.

procedure Refine($\{X_1, \ldots, X_n\}, Y, \Sigma$)
  Set $\mathbb{Y}$ to $\{Y\}$
  while There exist $Z \in \mathbb{Y}$, $Z' \subset Z$ and $A \in Z \setminus Z'$ such that $\Sigma \models Z' \to A$ **do**
    Remove Z from $\mathbb{Y}$
    if There is no $i \in \{1, \ldots, n\}$ such that $(Z' \cup \{A\}) \subseteq X_i$ **then**
      Add $Z' \cup \{A\}$ to $\mathbb{Y}$
    end if
    if There is no $i \in \{1, \ldots, n\}$ such that $(Z \setminus \{A\}) \subseteq X_i$ **then**
      Add $Z \setminus \{A\}$ to $\mathbb{Y}$
    end if
  end while
  return $\mathbb{Y}$
end procedure

The following example shows the benefit of splitting a complement into smaller pieces w.r.t. the propagation of updates.

Example 1. Let R be on $ABCDE$, let Σ consist of the FDs $B \to C$ and $D \to E$, and consider the view f_{ABD}. It can be checked that f_{BCDE} and $f_{BC,DE}$ are both complements of f_{ABD}. Let I be the instance such that $R^I = \{(a, b, c, d, e), (a, b', c', d', e')\}$. The insertion of (a, b, d') into $\pi_{ABD}(I)$ can be propagated back only by inserting (a, b, c, d', e') into I, for which $\pi_{BC}(I)$ and $\pi_{DE}(I)$ remain unchanged, but $\pi_{BCDE}(I)$ is required to include (b, c, d', e'). Hence, under the constant complement principle, the insertion of (a, b, d') into $\pi_{ABD}(I)$ is *translatable* [7,3] w.r.t. $f_{BC,DE}$, whereas it is not w.r.t. f_{BCDE}.

Algorithm 2 refines a view complement consisting of a single projection by computing one that consists of multiple projections and, although not minimal in general, transfers less (or the same) information.

Theorem 3. *Let $f_{X_1,\ldots,X_n}$ and f_Y be complementary views and let $\{Y_1, \ldots, Y_m\}$ be the result of* Refine($\{X_1, \ldots, X_n\}, Y, \Sigma$) *according to Algorithm 2. Then, (1) $f_{Y_1,\ldots,Y_m}$ is a complement of $f_{X_1,\ldots,X_n}$; and (2) $f_{Y_1,\ldots,Y_m} \leq f_Y$.*

Intuitively, Algorithm 2 works as follows: first, the set Y of attributes is recursively split into smaller sets according to the FDs in Σ; then, all the redundant sets, contained in one of the input sets $X_1, \ldots, X_n$, are discarded. The two steps are combined into a single one for efficiency reasons, in order to discard redundant sets at an earlier stage and so avoid unnecessary splittings. In the situation of Example 1, Refine($\{ABD\}$,$BCDE$,Σ) returns the sets of attributes BC and DE as expected.

6 Conclusion

For views defined by a single projection, we have established an easy correspondence between the information transfer order and the subset-relationship between the sets of attributes onto which the projections are defined. For sets of projections, we have shown with our new Saturation algorithm in Section 4 how the sets of attributes may possibly be augmented without changing the information transfer. Theorem 2 can be seen as stating the *soundness* of the Saturation algorithm. We conjecture that the opposite direction in Theorem 2 also holds, which amounts to claiming the *completeness* of the Saturation algorithm. Proving this conjecture remains as an open problem for future work.

Similarly, the Refine complement algorithm in Section 5 aims at transforming a complement given by one or several projections into another set of projections with smaller information transfer. Actually, one could easily further strengthen the algorithm by first applying the Saturation algorithm to the sets $X_1, \ldots, X_n$, which possibly increases these sets and thus may allow the deletion of more sets by the if-statements in the Refine complement algorithm. We conjecture that the resulting algorithm would then be guaranteed to return a view defined by a set of projections which is minimal w.r.t. information transfer. Again, the proof of this claim remains as an open problem for future work.

References

1. Abiteboul, S., Hull, R., Vianu, V.: Foundations of Databases. Addison-Wesley (1995)
2. Arenas, M., Pérez, J., Reutter, J.L., Riveros, C.: Foundations of schema mapping management. In: Proc. PODS 2010. pp. 227–238. ACM (2010)
3. Bancilhon, F., Spyratos, N.: Update semantics of relational views. ACM Trans. Database Syst. 6(4), 557–575 (1981)
4. Buneman, P., Khanna, S., Tan, W.C.: On propagation of deletions and annotations through views. In: Proc. PODS 2002. pp. 150–158. ACM (2002)
5. Caroprese, L., Trubitsyna, I., Truszczynski, M., Zumpano, E.: The view-update problem for indefinite databases. In: Proc. JELIA 2012. LNCS, vol. 7519, pp. 134–146. Springer (2012)
6. Cosmadakis, S.S., Papadimitriou, C.H.: Updates of relational views. J. ACM 31(4), 742–760 (1984)
7. Franconi, E., Guagliardo, P.: On the translatability of view updates. In: Proc. AMW 2012. CEUR WS Proceedings, vol. 866, pp. 154–167. CEUR-WS.org (2012)
8. Kimelfeld, B.: A dichotomy in the complexity of deletion propagation with functional dependencies. In: Proc. PODS 2012. pp. 191–202. ACM (2012)
9. Kimelfeld, B., Vondrák, J., Williams, R.: Maximizing conjunctive views in deletion propagation. ACM Trans. Database Syst. 37(4), 24 (2012)