

Technical Note

Comments on Bancilhon and Spyratos' "Update Semantics and Relational Views"

ARTHUR M. KELLER

The University of Texas at Austin

1. INTRODUCTION

Bancilhon and Spyratos [1] show how the choice of a complementary view will select a view update translator. They claim that "User requirements impose [the constraint that] . . . (ii) The user must have the means to cancel, if he wishes, the effect of every update that he is allowed on the view." (p. 562) If this requirement is eliminated, there are other reasonable translations that become acceptable. We present a particular view update translator that is quite reasonable, but that does not preserve any complement, and for which the requirement does not hold.

2. DEFINITIONS

Definition [1]. Let f and g be two functions whose domain is D . Then f and g are complementary mappings if

$$[\exists x, y \in D][(x \neq y) \wedge f(x) = f(y) \rightarrow g(x) \neq g(y)].$$

COROLLARY. Given a database D , a view v , and a complementary view c , there is at most one database state that corresponds to a desired view state (range of v) for a fixed view state (range of c).

The import of this corollary is that a view update translator that holds a complement constant has at most one translation. There are, however, view

This work was supported in part by contract N00039-82-G-0250 (the Knowledge Base Management Systems Project, Prof. G. Wiederhold, Principal Investigator) from the Defense Advanced Research Projects Agency and by contract AFOSR-80-0212 (Universal Relations, Prof. J. Ullman, Principal Investigator) from the Air Force Office of Scientific Research, both of the United States Department of Defense. The views and conclusions contained in this document are those of the authors and should not be interpreted as representative of the official policies of DARPA or the U.S. Government. This work was done primarily while the author was at the Computer Science Department of Stanford University.

Author's current address: The University of Texas at Austin, Department of Computer Sciences, Austin, TX 78712-1188.

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.

© 1987 ACM 0362-5915/87/0900-0521 \$01.50

update translators that have at most one translation that do not hold any complement constant. In the next section, we will illustrate a reasonable one.

3. A VIEW UPDATE TRANSLATOR

Consider the relation AB , with two attributes A and B , and the functional dependency $A \rightarrow B$. Let the domain of A contain at least one element, a_1 , and the domain of B contain at least two elements, b_1 and b_2 . We define the view V to select all tuples from AB where $B = b_1$.

We shall define a view update translator that accepts all single tuple updates valid in the view.

Insert tuple $\langle a, b \rangle$: If there exists a tuple $\langle a, y \rangle$, then replace $\langle a, y \rangle$ with $\langle a, b \rangle$, otherwise insert $\langle a, b \rangle$.

Delete tuple $\langle a, b \rangle$: Delete tuple $\langle a, b \rangle$ from the underlying database.

Replace tuple $\langle a, b \rangle$ by tuple $\langle c, d \rangle$: Perform translation for deleting $\langle a, b \rangle$ followed by translation for inserting $\langle c, d \rangle$.

Let us consider the translations of the insertion of the tuple $\langle a_1, b_1 \rangle$ starting with two different database states using this view update translator.

Initial database state 1:

A	B
a_1	b_2

Initial view state 1:

A	B
(empty relation)	

Result database state 1:

A	B
a_1	b_1

Result view state 1:

A	B
a_1	b_1

Initial database state 2:

A	B
(empty relation)	

Initial view state 2:

A	B
(empty relation)	

Result database state 2:

A	B
a_1	b_1

Result view state 2:

A	B
a1	b1

We observe that initial view state 1 and initial view state 2 are the same, yet initial database state 1 and initial database state 2 are different. Therefore, any complement view must have different values for initial database state 1 and initial database state 2. However, the result database states are the same. Thus, the result complement states must be the same. Consequently, the complement cannot remain constant.

If we wanted to hold constant a complement, we could, for example, choose the complement formed by selecting all tuples with $B \neq b1$. This would preclude accepting the insertion request above for database state 1. We could define another translator that holds another complement constant, but it could not implement all of these requests in the same way.

4. CONCLUSION

While view complements provide insight into the process of view update translation, requiring that a complement be chosen that remains constant may be too restrictive. Translators based on constant complements often do not translate all translatable database updates, and reject the remainder. When it is desirable to accept all translatable updates, this can restrict the choice of complements [2], or the only alternative may be a translator that does not preserve any complement. However, such a translator may not return the database back to its original database state after a series of view updates that return the view back to the original view state.

REFERENCES

1. BANCILHON, F., AND SPYRATOS, N. Update semantics and relational views. *ACM Trans. Database Syst.* 6, 4 (Dec. 1981), 557-575.
2. KELLER, A. M., AND ULLMAN, J. D. On complementary and independent mappings in the databases. In *the 1984 ACM/SIGMOD International Conference on Management of Data* (Boston, June 1984), ACM, New York, 1984, 143-148.