Some Open Problems Related to the Implied Constraint Problem

Stephen J. Hegner
Department of Computer Science and Electrical Engineering
Votey Building
University of Vermont
Burlington, VT 05405
U. S. A.

Phone: (802)656-3330
CSnet: hegner%uvm.edu@csnet-relay
USEnet: ..!uvm-gen!hegner

Address through August 1990:

Universitetet i Oslo
Matematisk Institutt
Postboks 1053 Blindern
0316 Oslo 3
NORWAY

Phone: +47-2-45-58-88
Net: m_hegner_s@use.uio.uninett

In this note I will identify several open problems of interest to me on the general topic of identifying the constraints on a view of a database schema. Informally, given a base schema and a view mapping, we wish to characterize exactly those states of the view schema which can arise as the image of a state of the base schema. More formally, we may proceed as follows.

A *database schema* is a pair $\mathbf{D} = (\mathbf{R}, \mathbf{C})$, with $\mathbf{R}$ a finite first-order language and $\mathbf{C}$ a set of sentences in the language of $\mathbf{R}$. In the traditional relational setting, $\mathbf{R}$ is just the set of relation names of the schema and $\mathbf{C}$ is the set of constraints. However, we do not explicitly disallow function symbols in $\mathbf{R}$; all of the problems given here make equal sense with function symbols allowed. We let $\mathsf{DB}(\mathbf{D}) = \mathsf{DB}(\mathbf{R})$ denote the set of all structures of the language of $\mathbf{R}$ (the *databases* of $\mathbf{D}$) and we let $\mathsf{LDB}(\mathbf{D})$ denote the set of all models of $\mathbf{C}$ (the *legal databases* of $\mathbf{D}$).

If $\mathbf{D_1} = (\mathbf{R_1}, \mathbf{C_1})$ and $\mathbf{D_2} = (\mathbf{R_2}, \mathbf{C_2})$ are database schemata, a *database mapping* $\gamma : \mathbf{D_1} \to \mathbf{D_2}$ is just an interpretation of the language of $\mathbf{R_2}$ into the language of $\mathbf{R_1}$, in the sense given in a mathematical context in [**?**] and in a database context in [**?**]. (This just amounts to expressing γ as a query in the relational calculus.) Such an interpretation induces in a natural way a mapping $\gamma^* : \mathsf{DB}(\mathbf{D_1}) \to \mathsf{DB}(\mathbf{D_2})$. We call γ *correct* if $\gamma^*(\mathsf{LDB}(\mathbf{D_1})) \subseteq \mathsf{LDB}(\mathbf{D_2})$, and we call γ a *view* if $\gamma^*(\mathsf{LDB}(\mathbf{D_1})) = \mathsf{LDB}(\mathbf{D_2})$.

We consider problems in which we are given a schema $\mathbf{D_1} = (\mathbf{R_1}, \mathbf{C_1})$ (the *base schema*), a second finite first-order language $\mathbf{R_2}$, (the *view language*), and an interpretation γ of the language of $\mathbf{R_2}$ in the language of $\mathbf{R_1}$. We are interested in identifying various ways of finding and/or testing a set of sentences Φ such that $\gamma : \mathbf{D_1} \to (\mathbf{R_2}, \Phi)$ is correct and/or a view. In other words, we seek to determine the constraints $\mathbf{C_2} = \Phi$ on the "view" schema $\mathbf{D_2} = (\mathbf{R_2}, \mathbf{C_2})$, implied, through γ, by the constraints on the base schema $\mathbf{D_1}$. We identify four specific problems. Except for the work of Klug [**?**] and Jacobs, Aronson, and Klug [**?**] on the testing problem, I am not aware of any work at all on these problems.

The Testing Problem This is the weakest form of the implied constraint problem which we consider. We are given a single sentence φ in the language of $\mathbf{R_2}$, and we ask whether or not $\gamma : \mathbf{D_1} \to (\mathbf{R_2}, \{\varphi\})$ is correct. In general, this problem is undecidable. (Just take $\mathbf{R_1} = \mathbf{R_2}$, γ = identity, and we are asking whether $\mathbf{C_1} \models \varphi$ in the language of $\mathbf{R_1}$; the implication problem for first-order logic). However, the question is semi-decidable; if $\gamma : \mathbf{D_1} \to (\mathbf{R_2}, \{\varphi\})$ is correct, we can detect this. The proof, as well as much further discussion on this issue, may be found in [**?**]. To achieve decidability, we must enforce some restrictions; the open problem is to determine useful ones. In other words, for "interesting" classes of database schemata and mappings, establish necessary and/or sufficient conditions for the testing problem to be decidable.

The Finiteness Problem In this problem, we assume that $\mathbf{C_1}$ is a finite set, and we ask if there is a *finite* Φ such that $\gamma : \mathbf{D_1} \to (\mathbf{R_2}, \Phi)$ is a view. We note that it is quite easy to produce very simple examples such that no such finite Φ exists. For example, take $\mathbf{R_1} = \{R[ABCD]\}$, $\mathbf{C_1} = \{B \to D,\ C \to D,\ DA \to B\}$, $\mathbf{R_2} = \{R[ABC]\}$, and $\gamma = \pi_{ABC}$.

The problem of identifying conditions under which there exists such a finite Φ is therefore nontrivial, even in very simple and practical settings.

The Axiomatiziability Problem Here we continue to assume that $\mathbf{C_1}$ is finite, and ask if Φ exists at all as a set of first-order sentences It is not difficult to construct simple examples in which $\mathbf{C_1}$ is finite, and yet no first-order Φ exists. For example, take $\mathbf{R_1} = \{R[AB]\}$, $\mathbf{C_1} = \{A \rightarrow B,\ B \rightarrow A\}$, $\mathbf{R_2} = \{R[A], R[B]\}$, and $\gamma = (\pi_A, \pi_B)$. Then Φ must express the constraint that the cardinality of the instance of $R[A]$ is the same as that of $R[B]$. This is not first-order in general, although it is when restricted to at-most-countable databases. As a general open problem, we seek necessary and or sufficient conditions such that the view is first-order axiomatizable. As a more specific open problem, we ask if there is a nice natural example of a view which is not first-order, even when attention is restricted to databases which are at most countable.

The Algorithmic Problem Here we ask specifically how to compute Φ effectively. More to the point, identify important special cases of base schemata and database mappings such that the axiomatization of the view is recursive. As a related open problem, we may ask also if it is possible to find cases in which the computation is also tractable (polynomial time).

Variations for Logic Databases In addition to the traditional framework of relational databases, this general problem also has application to logic databases. Within the latter framework, we regard $\mathbf{C_1}$ as the set of sentences comprising the "base" logic database, and we ask how to compute a set of sentences Φ which define the "view" logic database. The additional complication that arises from this point of view is that $\mathbf{C_1}$ will now vary as the base schema is updated. We must therefore additionally characterize admissible *families* of constraints $\mathbf{C_1}$ so that Φ is effectively computable for *each* member of this family.

References

[Ende72] Enderton, H. B. *A Mathematical Introduction to Logic*, Academic Press, 1972.

[JaAK82] Jacobs, B. E. , A. R. Aronson, and A. C. Klug, "On interpretations of relational languages and solutions to the implied constraint problem," *ACM TODS*, **7**,2(1982), pp. 291-315.

[Klug82] A. C. Klug, "Calculating constraints on relational expressions," *ACM TODS*, **5**, (1980), pp. 260-290.