CS632 Prelim IB

A. Demers

3 Apr 2003 – due 10 April 2003

At last here is the second part of the take-home. As usual, you may consult others for ideas and proof approaches, but please *reference your sources* and write up answers independently.

3 Vertical Decompositions

In this exercise we extend our treatment of vertical decompositions and adequacy of representation.

We start with a database schema containing a single relation, together with some integrity constraints:

$$\boldsymbol{R} = (R(U)) \qquad \text{and} \qquad C_R$$

Relation R is over attributes U. For this exercise the set C_R of integrity constraints may contain functional dependencies (FDs) each of which applies to a particular S_i. It may also contain key dependencies (KDs):

$$\text{Key}(X) \quad \equiv \quad X \rightarrow U$$

but these are just a special case of FDs.

We let $I(R)$ be the set of all instances of $\boldsymbol{R}$, whether or not they satisfy C_R; and $L(R)$ be the set of *legal* instances, which are required to satisfy C_R.

We next introduce $\boldsymbol{S}$ and its constraints:

$$\boldsymbol{S} = (S_1(U_1), \ldots, S_n(U_n)) \qquad \text{and} \qquad C_S$$

Here $\boldsymbol{S}$ has relations $S_i(U_i)$ satisfying

$$U \ = \ U_1 \cup U_2 \cup \ldots \cup U_n$$

The set C_S of integrity constraints may contain FDs and KDs, each of which applies to a particular S_i.

As above, $I(S)$ is the set of all instances of $\boldsymbol{S}$, whether or not they satisfy C_S; and $L(S)$ is the set of instances that satisfy C_S.

We use the standard transformation between $\boldsymbol{R}$ and $\boldsymbol{S}$:

$$\Pi_S : I(R) \rightarrow I(S) \qquad \text{and} \qquad \bowtie_S : I(S) \rightarrow I(R)$$

these are respectively projection onto the attributes in U_i and (natural) join.

Recall the first two of our definitions of adequacy:

Def: Decomposition $\boldsymbol{S}$ is α-adequate if

$$\Pi_S|_{L(R)} \ \leftrightarrow \ (\Pi_S(I(R)) \cup L(S))$$

That is, Π_S is a bijection between the specified sets. Note that Π_S need not be 1-to-1 when considered as a function on *all* of $I(R)$, only when restricted to $L(R)$ as above.

Def: Decomposition $\boldsymbol{S}$ is β-adequate if it is α-adequate and

$$\Pi_S(I(R) - L(R)) \ \subseteq \ (I(S) - L(S))$$

That is, in addition to being α-adequate, Π_S maps inconsistent instances to inconsistent instances.

In practice, we would like to be able to store the database according to the decomposed scheme S, so we want to know that enforcing the constraints in C_S is sufficient to guarantee consistency with respect to the original schema; that is, satisfying C_S should guarantee that C_R would be satisfied as well.

So here is yet another adequacy definition.

Def: Decomposition $\boldsymbol{S}$ is ϕ-adequate if

$$\Pi_S|_{L(R)} \ \leftrightarrow \ (L(S))$$

that is, if Π_S is a 1-to-1 correspondence between consistent instances in $L(R)$ and $L(S)$.

Question 2a: We know that β-adequacy implies α-adequacy. What is the relation between β-adequacy and ϕ-adequacy? In each direction, either prove the implication or exhibit a counterexample.

We now explore what happens when C_S may contain interrelational constraints in the form of inclusion dependencies (IDs). Recall an ID has the form

$$S_i[A_1, \ldots, A_k] \;\subseteq\; S_j[B_1, \ldots, B_k]$$

and is satisfied if

$$(\forall t \in s_i)(\exists t' \in s_j)\; t[A_1, \ldots, A_k] \;=\; t'[A_1, \ldots, A_k]$$

Obviously an ID may be either an interrelational or intrarelational constraint, depending on whether S_i and S_j are the same. An interrelational ID is often called a *foreign key.*

Question 2b: (An aside). When discussing BCNF, which essentially reduces FDs to KDs, we argued that this was a good thing to do in practice because a database management system necessarily has a B-tree indexing implementation (or something equivalent), and it is possible to implement KDs efficiently using that functionality. Give a similar argument for IDs / foreign keys.

Here is yet another normal form:

Def: Scheme S is in *IK normal form* if C_S consists only of (intrarelational) KDs and (interrelational) IDs.

Question 2c: Describe an algorithm that, starting with a schema $\boldsymbol{R} = (R(U))$ and a set C_R of FDs, finds a ϕ-adequate decomposition $(\boldsymbol{S}, C_S)$ in IK normal form.

Is the resulting decomposition β-adequate as well?

4 Serializability

In this question we will not consider multi-version schedules.

Recall that for unrestricted transactions conflict serializability (CSR) is strictly stronger than view serializability (VSR). That is, every CSR schedule is necessarily VSR, but there exist VSR schedules that are not CSR. (Actually, we showed this for transactions satisfying the requirement that no entity x is read or written more than once.)

For some more constrained transaction models this distinction disappears. Recall in the *restricted model* (which prohibits blind writes by requiring that every entity x written in a transaction must have been read by some earlier step of the transaction) a schedule is VSR if and only if it is CSR.

For each of the following models, either give a schedule that is VSR but not CSR, or else argue that no such schedule exists.

(3a) the *very short transaction* model: each transaction consists of a single (read or write) step.

(3b) The *short transaction* model: each transaction consists of at most 2 steps.

(3c) The *atomic write* model: each transaction consists of a sequence of 0 or more read steps, followed by a single write step that writes the values of 0 or more entities atomically.

(3d) The *single write* model: each transaction contains at most one write step (which writes a single entity).

(3e) The *review* model: if an entity x is written by a transaction, then x must be read by some later step of the same transaction (a transaction is allowed to read the same entity more than once).