Theoretical Computer Science 19 (1982) 267–285
North-Holland Publishing Company

INCLUSION AND EQUIVALENCE BETWEEN RELATIONAL DATABASE SCHEMATA

Paolo ATZENI, Giorgio AUSIELLO and Carlo BATINI
Istituto di Automatica, Università di Roma, Roma, Italy

Marina MOSCARINI
Istituto di Analisi dei Sistemi ed Informatica, CNR, Roma, Italy

Communicated by Z. Pawlak

Abstract. Conceptual relations among relational database schemata are investigated.

Two different definitions of inclusion and a definition of equivalence between schemata are given.

Several examples of practical situations adequately handled by our definitions are shown.

Finally, necessary and sufficient conditions for inclusion and equivalence are proved for two classes of schemata, meaningful in the relational theory.

1. Introduction and motivation

In database design methodologies several steps exist that involve transformations of the database schema according to specific needs.

Examples of such transformations are:
- the integration of user views into a single global schema;
- in case of distributed databases, the distribution of the global schema over a set of nodes;
- the generation of views from the global schema;
- the decomposition or composition of elements of the schema (e.g. record types in the network model, relations in the relational model) for the needs of both logical (e.g. normalizations) and physical (optimization of access paths) design.

For each of such transformations a particular relationship must hold between the original and the transformed schemata. For example:
- the distributed schema must embody exactly the same information as the original global schema;
- each view may only represent information that can also be represented in the global schema.

In order to formalize these comparisons and relationships there is the need of precise definitions of *conceptual relations* among database schemata, that is, relations that compare the ability of database schemata to represent information and answer to queries.

In this paper various conceptual relations among relational database schemata are proposed. Furthermore they are studied for a sufficiently general class of schemata, particularly meaningful in the relational theory.

In the relational literature various contributions exist that treat conceptual relations among schemata, but they are seldom comparable, because:
- sometimes there is no formal definition;
- different assumptions are made on the specific relational model that is adopted.

The first conceptual relations for relational databases were proposed by Codd in his fundamental paper of 1972 on normalization [8]. In a framework in which the only type of integrity constraint that is allowed is the functional dependency, two kinds of equivalence among schemata:
- the "insertion deletion equivalence",
- the "query-equivalence",

were defined and studied between schemata obtained the one from the other via normalization.

From an informal point of view, the "insertion–deletion" equivalence between schemata corresponds to the possibility of performing on them exactly the same operations of insertion and deletion of pieces of information, while the "query-equivalence" corresponds to the possibility of extracting from them, by means of queries, exactly the same information. Codd shows how, in his framework, an unnormalized schema and its corresponding normalized schema are neither insertion–deletion equivalent (because the normalized schema allows more kinds of insertions and deletions than the unnormalized one) nor query equivalent (because the set of admissible states for the unnormalized schema is query-equivalent to a proper subset of the set of admissible states for the normalized schema).

Since Codd's paper many authors have studied decomposition and normalization, proposing different approaches (see [7, 11, 15]) that, as a consequence, lead to different conceptual relations between the unnormalized and the corresponding normalized schemata.

All this work was surveyed and its formalism clarified in a paper presented in 1978 by Beeri, Bernstein and Goodman [5]. There, under the strong "universal relation assumption" (i.e., for each admissible state of the database of interest, all relations are projections of a single relation), three (actually, four, but we are not interested in the first one) conceptual relations between a monorelational schema $S_\emptyset$ and a decomposed schema S_D were proposed:
- S_D Rep2 $S_\emptyset$ if they have the same attributes and the same data dependencies;
- S_D Rep3 $S_\emptyset$ if they have the same attributes and the states of S_D contain the same data as the states of $S_\emptyset$;
- S_D Rep4 $S_\emptyset$ if both S_D Rep2 $S_\emptyset$ and S_D Rep3 $S_\emptyset$ hold (and, as a consequence, there is a one to one mapping between the states of $S_\emptyset$ and the states of S_D).

Then, it was shown which of these relations hold between the unnormalized and the normalized schemata in the various approaches.

In the same framework of universal relation assumption and projection–join transformation, a more formal approach to schema equivalence was proposed by Beeri, Mendelzon, Sagiv and Ullman in [6], based on "the equivalence of the sets of fixed points of the project–join mapping associated with the database schemes in question".

All the previous papers concern conceptual relations between schemata obtained the one from the other by means of vertical decompositions and compositions, that is, transformations based on projection and join operators. On the other hand several authors have stressed the importance of other kinds of transformations. For instance, Fagin [12] and Sciore [16] have devoted attention to "horizontal" decompositions and compositions, i.e., obtained by use of restriction and union operators. In the framework of normalization, such decompositions are useful for schemata in which hidden functional dependencies are involved (see [17]). They are also meaningful in distributed database design.

More recently, Kandzia and Klein in [14] and the authors in [1, 2, 3, 4] have proposed a more general approach in which all restrictions were removed.

The main characteristics of our approach are:

- to base the comparison of database schemata on the ability to answer queries;
- to consider not only vertical transformations but also horizontal and even more general ones;
- to establish a framework in which database schemata may be compared both when the existence of the universal relation is assumed and when it is not;
- to allow the comparison of database schemata in which any kind of integrity constraint can be defined.

Due to the generality of the approach the conceptual relations defined in these papers (and in the present one) allow, at least in principle, to deal with all the situations described at the beginning of this section.

The paper is organized as follows:

In Section 2, two different types of conceptual inclusion and equivalence between schemata are defined, based on the above criteria. Such concepts are defined in terms of a query language Q and an integrity constraints language IC. In Section 2 we choose not to fix such languages in order to show that in a wide generality the existence of a conceptual relation between two schemata may depend upon Q and IC.

In Section 3 several examples of pairs of schemata are shown, with the corresponding conceptual relation.

In Section 4 we choose for Q the relational algebra and for IC tuple predicates and functional dependencies, and we show that in this formal framework it is possible to formally characterize in terms of decidable properties the above defined conceptual relations for a sufficiently general class of schemata, meaningful in the relational database theory.

In Section 5 further research developments are described.

2. Basic definitions

As we already stated in the introduction, throughout this paper we will study inclusion and equivalence of database schemata with respect to their conceptual content, by referring to a specific data model, precisely the n-ary relational model (see [8, 9, 10]).

Definition 2.1. A *database schema* (dbs) is a 3-tuple $S = \langle A, R, V\rangle$ where
- A is a finite set of names of *attributes* $\{A1, A2, \ldots, An\}$; to each Ai a domain of values Di is associated;
- R is a m-tuple of *relations*, each composed of a relation name and a set of attributes: $\langle R1(A11, \ldots, A1h_1), \ldots, Rm(Am1, \ldots, Amh_m)\rangle$.
- V is the set of *integrity constraints*.

Definition 2.2. *A tuple of a relation* $Rj(Aj1, Aj2, \ldots, Ajh_j)$ is a set of pairs $\{(Aj1 : xj1), \ldots, (Ajh_j : xjh_j)\}$ such that $\forall i, xji \in Dji$.

Definition 2.3. *Instance of a relation* $Rj(Aj1, Aj2, \ldots, Ajh_j)$ is a set of tuples of relation Rj. In the following $\overline{Rj}$ will denote an instance of Rj.

Definition 2.4. Given a schema $S = \langle A, R, V\rangle$ with $R = \langle R1, R2, \ldots, Rm\rangle$, *instance of* S is an m-tuple $i = \langle \overline{R1}, \overline{R2}, \ldots, \overline{Rm}\rangle$ such that:
- $\forall 1 \leq j \leq m$, $\overline{Rj}$ is an instance of $Rj(Aj1, Aj2, \ldots, Ajh_j)$,
- i satisfies the integrity constraints.

In the following I is the set of instances of a schema S.

Example 2.1. Let us consider the schema PAYROLL concerning the employees of an enterprise and their salaries:

$$\text{PAYROLL} = \langle A, R, V\rangle$$

where
- R contains a unique relation Employee(Emp#, Salary),
- the domain of Emp# is the "set of integers from 00001 to 99999",
- the domain of Salary is the "set of integers from 10^5 to 10^8",
- V contains the following constraint: "every employee has exactly one salary" (this is a functional dependency and corresponds to usual notation Emp# $\rightarrow$ Salary).

Now, we can introduce our concepts of inclusion and equivalence between schemata.

Intuitively, two schemata S, S' are equivalent if for each instance i of S an instance i' of S' exists from which we can extract exactly the same information and vice versa. This latter concept may be formalized saying that for each query q on i a query q' on i' must exist such that they give exactly the same answer. In [9] and [2] it has been shown that this condition holds if and only if a query on i exists whose result is i' and a query on i' exists whose result is i. Our definitions are based on this last property.

With regard to the inclusion of schemata, we may be interested in two kinds of situations:

- for each instance i of s an instance i' of s' exists that contains at least the same information;
- for each instance i of s an instance i' of s' exists that contains exactly the same information.

These two situations arise respectively when we consider a view with respect to the global schema and when we ask that a decomposed schema does not loose any information.

As a consequence we give two different definitions of inclusion between schemata.

We use the following notations:

- given a query language Q, Q^+ indicates the set of tuples of elements of Q;
- given $\bar{f}=\langle f_1, f_2, \ldots, f_n\rangle \in Q^+$, $i_1=\langle \bar{R}_{11}, \bar{R}_{12}, \ldots, \bar{R}_{1n}\rangle$ and i_2, $i_1=\bar{f}(i_2)$ indicates that $\bar{R}_{1j}=f_j(i_2)\ \forall 1 \leq j \leq n$.

Definition 2.5. Let $S1$ and $S2$ be two schemata.

(1) $S1$ is weakly *included* in $S2$ ($S1 < S2$) (with respect to the query language Q) if $\bar{f} \in Q^+$ exists such that for every instance $i1 \in I1$ an instance $i2 \in I2$ exists such that $i1=\bar{f}(i2)$.

(2) $S1$ is *included* in $S2$ ($S1 \sqsubset S2$) (with respect to the query language Q) if $\bar{f}, \bar{f}' \in Q^+$ exist such that for every instance $i1 \in I1$ an instance $i2 \in I2$ exists such that $i1=\bar{f}(i2)$ and $i2=\bar{f}'(i1)$.

(3) $S1$ *is equivalent to* $S2$ ($S1 \sqsubseteq\!\!\!\sqsupseteq S2$) (with respect to the query language Q) if $S1 \sqsubset S2$ and $S2 \sqsubset S1$ (with respect to Q).

It is easy to show that the relation of equivalence between database schemata satisfies the properties of reflexivity, symmetry and transitivity and so it actually is an equivalence relation.

3. Applications

Before introducing the main results that characterize the properties of the concepts of inclusion and equivalence according to our definitions, we show that the given concepts are adequate to describe the situations that were discussed in the introduction.

Views. As we observed in the preceding section, in the case of views we are interested in establishing a correspondence between instances such that for each instance of the view there is an instance of the global schema that provides at least the same information. This situation is clearly modelled by the weak inclusion.

Example 3.1.

Emp(Emp#, Name, Salary, Address) Emp# → Name, Salary, Address	Payroll(Emp#, Salary) Emp# → Salary
$S1$	$S2$ (View on $S1$)

Clearly, $S2 < S1$.

Lossless (vertical) decomposition. In this case the required correspondence is that for each instance of the undecomposed schema there exists an instance of the decomposed schema which contains exactly the same information (but, in general, not viceversa).

Example 3.2.

Emp (Emp#, City, ZIP) Emp# → City City → ZIP	Emp1 (Emp#, City) Emp# → City Emp2 (Emp#, ZIP) Emp# → ZIP
$S1$	$S2$

In this case we have $S1 \sqsubseteq S2$.

Dependency preserving (vertical) decomposition. In the following example we show that when the decomposition preserves the dependencies (this is the property, for instance, guaranteed by Bernstein's normalization [7] algorithm under the Universal Relation Assumption[1]), the decomposed schema is included in the undecomposed one.

Example 3.3.

R (Emp#, Dept, Project) Emp# → Dept Project → Dept	$R1$ (Emp#, Dept) Emp# → Dept $R2$ (Project, Dept) Project → Dept $\forall\langle\overline{R1}, \overline{R2}\rangle$ PROJ($\overline{R1}$; Dept) = PROJ($\overline{R2}$; Dept)
$S1$	$S2$

[1] In this case [13] the Universal Relation Assumption corresponds to the presence, in the decomposed schema, of the constraint of equality of the projections:

$\forall\langle\overline{R1}, \overline{R2}\rangle$ (PROJ($\overline{R1}$; Dept) = PROJ($\overline{R2}$; Dept)).

$S2 \sqsubset S1$. In fact the instances in $S1$ that do not satisfy the lossless join property have no counterpart in $S2$; while each instance in $S2$ has a counterpart in $S1$.

Independent (vertical) decomposition. In this case, studied in [16] the decomposition is both lossless and dependency preserving. Furthermore in the decomposed schema the constraint of the existence of the Universal Relation is also assumed.

Example 3.4.

$S1$	$S2$
Emp (Emp#, City, ZIP) Emp# → City City → ZIP	Emp (Emp#, City) Emp# → City Address (City, ZIP) City → ZIP $\forall\langle\overline{R1}, \overline{R2}\rangle$ PROJ ($\overline{R1}$; City) = PROJ ($\overline{R2}$; City)

Here we have $S1 \mathrel{\underline{\sqsubset}} S2$.

All the above examples refer to vertical decompositions. The following one shows how our theory can handle also more general kinds of transformations.

Horizontal decomposition. During the design of a database (especially of a distributed one) a horizontal decomposition of a relation is often needed. Usually, it is required that this decomposition be performed without loss of information.

Example 3.5.

$S1$ (undecomposed schema)	$S2$ (decomposed schema)
$R1$ (SocSecNo, Age) SocSecNo → Age	$R1$ (SocSecNo, Age) SocSecNo → Age $\forall \overline{R1}\ \forall t \in \overline{R1}\ t.\text{Age} \leqslant 18$ $R2$ (Soc SecNo, Age) SocSecNo → Age $\forall \overline{R1}\ \forall t \in \overline{R1}\ t.\text{Age} \geqslant 6$

In this case, $S1 \sqsubset S2$, since the instances of $S2$, such that information about some people between 6 and 18 is stored in one relation only, have no counterpart in $S1$.

In distributed databases, in order to split and distribute a global schema into a set of local schemata it is required that the set of local schemata embody exactly the same information as the global schema.

So, to perform horizontal decomposition in a framework of distributed databases we need to add the following constraint to $S2$.

$$\forall\langle\overline{R1}, \overline{R2}\rangle\ \mathrm{RESTR}(\overline{R1}, \mathrm{Age} \geq 6) = \mathrm{RESTR}(\overline{R2}, \mathrm{Age} \leq 18)$$

that requires that information about each person whose age is between 6 and 18 is stored in both files in order to have $S1 \subsetneqq S2$.

Hidden functional dependencies and multiple decomposition.

Example 3.6.

R (Project, Budget, Dept)
Project → Budget
if Budget < 30000 *then* Project → Dept

$S1$

$R1$ (Project, Budget, Dept)
Project → Budget, Dept
$\forall\overline{R1}\forall t \in \overline{R1}\ t.\mathrm{Budget} < 30.000$
$R2$ (Project, Budget)
Project → Budget
$\forall\overline{R2}\forall t \in \overline{R2}\ t.\mathrm{Budget} \geq 30.000$
$R3$ (Project, Dept)
$\forall\langle\overline{R1}, \overline{R2}, \overline{R3}\rangle$:
(1) $\mathrm{PROJ}(\overline{R2}; \mathrm{Project}) = \mathrm{PROJ}(\overline{R3}; \mathrm{Project})$
(2) $\mathrm{PROJ}(\overline{R1}; \mathrm{Project}) \cap \mathrm{PROJ}(\overline{R2}; \mathrm{Project}) = \emptyset$

$S2$

Here the reason for decomposition is that in $S1$ the f.d. Project → Dept does not hold in general but only for suitable restrictions (this is an example of hidden functional dependency [17]): this forces an horizontal decomposition, followed by a vertical one. Again we have $S1 \subsetneqq S2$.

4. Conceptual relations among schemata

4.1. *Introduction*

In this section we shall characterize conceptual relations among schemata for particular classes of schemata meaningful in the theory of relations, that is schemata obtained both by "vertical" and by "horizontal" decompositions and compositions.

With regard to the query language, we assume in the following a language in which the following operators are defined:

(1) UNION$(\overline{R1}, \overline{R2}, \ldots, \overline{Rn}) = \{t \mid (\exists 1 \leq j \leq n)\ (t \in \overline{Rj})\}$ where all the Rj are defined on the same set of attributes X; (such relations are said union-compatible [10]);

(2) RESTR$(\bar{R}; p) = \{t \mid t \in \bar{R} \wedge p(t)\}$ where p is a boolean expression, that we call in the following *tuple predicate*, whose terms may be: $t.A\theta c$ or $t.A\theta t.B$ where
- A and B are attributes of R,
- c belongs to the domain of A,
- $\theta \in \{=, \neq, <, >, \leq, \geq\}$.

(3) PROJ$(\bar{R}; X) = \{t.X \mid t \in \bar{R}\}$ where X is contained in the set of attributes of R and $t.X$ denotes the set of pairs $(Ai : xi) \in t$ such that $Ai \in X$;

(4) JOIN$(\overline{R1}, \overline{R2}) = \{t1 \cup t2 \mid t1 \in \overline{R1} \wedge t2 \in \overline{R2} \wedge t1.X = t2.X\}$ where X is the set of common attributes between $R1$ and $R2$.

In Section 4.2 we investigate conceptual relations among schemata obtained by "horizontal" decompositions (i.e. related by UNION and RESTR operators).

In Section 4.3 we investigate conceptual relations among schemata obtained by "vertical" decompositions (i.e. related by PROJ and JOIN operators).

4.2. *Conceptual relations among schemata related by restriction and union operators*

We consider initially a pair of schemata $S1$ and $S2$ where:
- $S1$ contains:
  - $R(X)$,
  - $V1 = \{p\}$ where p is a tuple predicate over R;
- $S2$ contains:
  - $R1(X), R2(X), \ldots, Rn(X)$,
  - $V2 = \{p1, p2, \ldots, pn\}$ where pi is a tuple predicate over Ri.

The following theorems characterize sufficient conditions for the existence of the conceptual relations defined in Section 2 for the above mentioned pair of schemata $S1$ and $S2$.

Theorem 4.2.1. *$(p1 \vee p2 \vee \cdots \vee pn \Rightarrow p) \wedge (\forall 1 \leq i \leq n)\quad (\forall 1 \leq j \leq n)\quad (i \neq j \Rightarrow pi \wedge pj = false) \Rightarrow S2 \sqsubset S1$.*

Proof. For any $i1 = \langle \bar{R} \rangle \in I1$ and $i2 = \langle \overline{R1}, \overline{R2}, \ldots, \overline{Rn} \rangle \in I2$ let $\bar{f}$, $\bar{f}'$ be the following:
- $\bar{f} = \langle f1 \rangle$ where $f1(i2) = $ UNION$(\overline{R1}, \overline{R2}, \ldots, \overline{Rn})$,
- $\bar{f}' = \langle f1', f2', \ldots, fn' \rangle$ where $fi'(i1) = $ RESTR$(\bar{R}; pi)$.

We have to prove that for every instance $i2 \in I2$ there exists an instance $i1 \in I1$ such that $i2 = \bar{f}'(i1)$ and $i1 = \bar{f}(i2)$. Given $i2 = \langle \overline{R1}, \overline{R2}, \ldots, \overline{Rn} \rangle \in I2$ let $i1 = \langle \bar{R} \rangle$ where $\bar{R} = $ UNION$(\overline{R1}, \overline{R2}, \ldots, \overline{Rn})$.

(1) $i1 \in I1$. In fact $\bar{R}$ verifies the tuple predicate p for hypothesis $p1 \vee p2 \vee \cdots \vee pn \Rightarrow p$.

(2) $i2 = \bar{f}'(i1)$. In fact:

$$(\forall 1 \leqslant j \leqslant n)\, fj'(i1) = \text{RESTR}(\bar{R}; pj)$$
$$= \text{RESTR}(\text{UNION}(\overline{R1}, \overline{R2}, \ldots, \overline{Rn}); pj) = \overline{Rj}$$

where the last equality derives from hypothesis $pj \wedge p = false$ for $j \neq k$.

(3) $i1 = \bar{f}(i2)$. In fact $\bar{R} = f1(i2)$ by definitions of $\bar{f}$ and $i2$. □

Theorem 4.2.2. $(p \Rightarrow p1 \vee p2 \vee \cdots \vee pn) \Rightarrow S1 \sqsubseteq S2$.

Proof. Let $\bar{f}$ and $\bar{f}'$ be as in Theorem 4.2.1. Given $i1 = \langle \bar{R} \rangle \in I1$ let $i2 = \langle \overline{R1}, \overline{R2}, \ldots, \overline{Rn} \rangle$ where $\overline{Rj} = \text{RESTR}(\bar{R}; pj) \forall 1 \leqslant j \leqslant n$.

(1) $i2 \in I2$. In fact each $\overline{Rj}$ satisfies, by definition, the corresponding tuple predicate pj.

(2) $i1 = \bar{f}(i2)$. In fact

$$f1(i2) = \text{UNION}(\overline{R1}, \overline{R2}, \ldots, \overline{Rn})$$
$$= \text{UNION}(\text{RESTR}(\bar{R}; p1), \text{RESTR}(\bar{R}; p2), \ldots, \text{RESTR}(\bar{R}; pn))$$
$$= \bar{R}$$

where the last equality derives from hypothesis $p \Rightarrow p1 \vee p2 \vee \cdots \vee pn$.

(3) $i2 = \bar{f}'(i1)$. In fact $\overline{Rj} = fj'(i1)\ \forall 1 \leqslant j \leqslant n$ by definitions of $\bar{f}'$ and $i2$. □

Corollary 4.2.1. $(p1 \vee p2 \vee \cdots \vee pn = p) \wedge (\forall 1 \leqslant i \leqslant n)\,(\forall 1 \leqslant j \leqslant n) \quad (i \neq j \Rightarrow pi \wedge pj = false)) \Rightarrow S1 \subseteqq S2$.

Proof. From Theorems 4.2.1 and 4.2.2. □

We can generalize Corollary 4.2.1 if we introduce the concept of Class of Union Compatibility (CUC). Given a schema S, we call CUC an equivalence class over the set of relations in S established by the Uniform Compatibility relation.

Theorem 4.2.3. *Let $S1$ and $S2$ be two schemata such that*:

- *$S1$ contains*:
  - *relations $R11, R12, \ldots, R1n$,*
  - *$V1 = \{p11, p12, \ldots, p1n\}$, a set of tuple predicates over such relations*;
- *$S2$ is similarly defined.*

If for every CUC $C1i$ in $S1$ a corresponding CUC $C2j$ in $S2$ exists such that

$$\bigvee_k p1k = \bigvee_h p2h$$

where $R1K \in C1i$ and $R2h \in C2j$ and vice versa, and $(\forall k \neq h)(\forall j)pjk \wedge pjh = false$, then $S1 \eqsim S2$.

Proof. Let $S3$ be a schema that contains for every CUC $C1i$ in $S1$ a relation Ri and $V3$ is a set of tuple predicates such that $pi = \bigvee_k pik$ where $R1k \in C1i$. It is easy to show that $S3 \eqsim S1$. In fact every pair of schemata:

(1) $S1i$ containing the set of relations in $C1i$ and the corresponding set of tuple predicates;

(2) $S3i$ containing the relation Ri and the tuple predicate pi

is equivalent according to Corollary 4.2.1.

Furthermore, juxtaposing pairwise equivalent schemata we still obtain a couple of equivalent schemata, so $S3 \eqsim S1$.

Analogously $S3 \eqsim S2$. For transitivity $S1 \eqsim S2$. □

In the former theorems we stated and proved which properties imply inclusion and equivalence relations among database schemata whose instances are obtained by means of transformations based on UNION and RESTR operators.

With the following theorems we show that the conditions expressed in Corollary 4.2.1 are also necessary when the query language is the language defined in Section 4.1.

In the proofs the following facts are used:

Fact 4.2.1. If $\overline{R1} = \overline{R2} = \cdots = \overline{Rn} = \emptyset$, the result of application of any operator 1–4 to $\overline{R1}, \overline{R2}, \ldots, \overline{Rn}$ is $\emptyset$.

Fact 4.2.2. If $(A : x) \in t$ and t is a tuple that belongs to the relation resulting from the application of any operator 1–4, then one of the operands must contain a tuple t' such that $(A : x) \in t'$.

Theorem 4.2.4. $S2 \sqsubset S1 \Rightarrow ((p1 \vee p2 \vee \cdots \vee pn \Rightarrow p) \wedge (\forall 1 \leq i \leq n) \quad (\forall 1 \leq j \leq n)$ $(i \neq j \Rightarrow pi \wedge pj) = false))$.

Proof. (1) $S2 \sqsubset S1 \Rightarrow (p1p2 \vee \cdots \vee pn)$. We proceed by contradiction. Suppose a tuple t exists such that

$$(p1 \vee p2 \vee \cdots \vee pn)(t) = true \wedge (p(t) = false).$$

Let $pk(t) = true$, with $1 \leq k \leq n$. For hypothesis we have $S2 < S1$; so $\bar{f}$, $\bar{f}'$ exist such that

$$(\forall i2 \in I2)(\exists i1 \in I1)(i2 = \bar{f}'(i1) \wedge i1 = \bar{f}(i2)).$$

Consider now the instance $i2 = \langle \overline{R1}, \overline{R2}, \ldots, \overline{Rn} \rangle \in I2$ with $\overline{Rk} = \{t\}$ and $\overline{Rj} = \emptyset$ for every $j \neq k$. For Fact 4.2.2 and hypothesis $p(t) = false$, the only possible instance $i1=\bar{f}(i2)$ is $i1=\langle\bar{R}\rangle$ with $\bar{R}=\emptyset$. But for Fact 4.2.1 no $\bar{f}'$ exists such that $i2=\bar{f}'(i1)$.

(2) $S2 \sqsubset S1 \Rightarrow (\forall 1 \leqslant i \leqslant n)(\forall 1 \leqslant j \leqslant n)(i \neq j \Rightarrow p_i \wedge p_j = false)$. Suppose, by contradiction, that i, j $(i \neq j)$ and a tuple t exist such that

$$pi(t)=pj(t)=true.$$

From $S2 \sqsubset S1$, $\bar{f}, \bar{f}'$ exist such that

$$(\forall i2 \in I2)(\exists i1 \in I1)(i2=\bar{f}'(i1) \wedge i1=\bar{f}(i2))$$

with $\bar{f}$ injective (trivial, by definition of inclusion).
Consider the following instances of $I2$:

$$\langle\overline{R1}=\emptyset, \overline{R2}=\emptyset, \ldots, \overline{Ri}=\{t\}, \ldots, \overline{Rn}=\emptyset\rangle,$$
$$\langle\overline{R1}=\emptyset, \overline{R2}=\emptyset, \ldots, \overline{Rj}=\{t\}, \ldots, \overline{Rn}=\emptyset\rangle,$$
$$\langle\overline{R1}=\emptyset, \overline{R2}=\emptyset, \ldots, \overline{Ri}=\{t\}, \ldots, \overline{Rj}=\{t\}, \ldots, \overline{Rn}=\emptyset\rangle,$$
$$\langle\overline{R1}=\emptyset, \overline{R2}=\emptyset, \ldots, \overline{Rn}=\emptyset\rangle.$$

By Fact 4.2.2 any mapping $\bar{f}$ may establish a correspondence from any of the four preceding instances of $I2$ to only one of the two following instances of $I1$:

$$\langle\bar{R}=\emptyset\rangle, \qquad \langle\bar{R}=\{t\}\rangle$$

and so $\bar{f}$ cannot be injective. □

Corollary 4.2.2. $S2 \sqsubset S1 \Leftrightarrow (((p1 \vee p2 \vee \cdots \vee pn) \Rightarrow p) \wedge ((\forall 1 \leqslant i \leqslant n)(\forall 1 \leqslant j \leqslant n)$ $(j \neq i \Rightarrow pi \wedge pj = false)))$.

Theorem 4.2.5. $S1 \sqsubset S2 \Rightarrow (p \Rightarrow p1 \vee p2 \vee \cdots \vee pn)$.

Proof. We proceed by contradiction. Suppose a tuple t exist such that

$$(p(t)=true) \wedge (p1 \vee p2 \vee \cdots \vee pn)(t)=false.$$

By hypothesis, two $\bar{f}, \bar{f}'$ exist such that

$$(\forall i1 \in I1)(\exists i2 \in I2)(i1=\bar{f}(i2) \wedge i2=\bar{f}'(i1)).$$

Consider now the instance $i1=\langle\bar{R}\rangle \in I1$ with $\bar{R}=\{t\}$. By Fact 4.2.2 and the hypothesis $(p1 \vee p2 \vee \cdots \vee pn)(t)=false$ the unique instance $i2 \in I2$ such that $i2=\bar{f}'(i1)$ is $i2=\langle\overline{R1}, \overline{R2}, \ldots, \overline{Rn}\rangle$ with $\overline{R1}=\overline{R2}=\cdots=\overline{Rn}=\emptyset$. By Fact 4.2.1 no $\bar{f}$ exists such that $i1=\bar{f}(i2)$. □

Corollary 4.2.3. $S1 \sqsubset S2 \Leftrightarrow (p=p1 \vee p2 \vee pn)$.

Corollary 4.2.4. $S1 \subsetneqq S2 \Leftrightarrow (p=p1 \vee p2 \vee \cdots \vee pn) \wedge (\forall 1 \leqslant i \leqslant n)(\forall 1 \leqslant j \leqslant n)$ $(i \neq j \Rightarrow pi \wedge pj = false)$.

4.3. *Conceptual relations among schemata related by join and projection operators*

Let $S1$ and $S2$ be two schemata such that
- $S1$ contains:
  - $R(X)$,
  - $V1 = \mathrm{FD1}$;
- $S2$ contains:
  - $R1(X1)$, $R2(X2)$,
  - $V2 = \mathrm{FD2}$

where X, $X1$, $X2$ are sets of attributes such that $X \supset X1 \cap X2 = X12 \neq \emptyset$, FD1 and FD2 are sets of functional dependencies (fds).

We introduce now the following properties:

A. $X1 \cup X2 \supseteq X$;

A'. $X1 \cup X2 \subseteq X$;

B'. $\mathrm{FD1}^+ \supseteq \mathrm{FD2}^+$;

B. $\mathrm{FD2}^+ \supseteq \mathrm{FD1}^+$;

C. $V2$ contains (besides FD2) the following constraint:

$$(\forall \langle \overline{R1}, \overline{R2} \rangle)(\mathrm{PROJ}(\overline{R1}; X12) = \mathrm{PROJ}(\overline{R2}; X12));$$

D. $X12 \to X1 \in \mathrm{FD1}^+ \vee X12 \to X2 \in \mathrm{FD1}^+$

in which $\mathrm{FD1}^+$ denotes the closure of the set of functional dependencies FD1 defined in [5] and $\mathrm{FD2}^+$ denotes the closure of the set of fds FD2 defined as

$$\mathrm{FD2} = \mathrm{FDR1} \cup \mathrm{FDR2}$$

where FDR1 and FDR2 are the sets of fds defined on $R1$ and $R2$. The definition of $\mathrm{FD2}^+$ can be easily extended to schemata with more than two relations.

Notice that in our approach we need a specific formal definition for the closure of the set of fds of a schema with more than one relation: this happens because we choose to follow an approach as general as possible to the study of conceptual relations among schemata. From that choice we do not suppose as valid the "Universal Relation Assumption", accepted for instance in [5, 6].

The following theorem points out that, if the following property holds:

E. the constraints language allows only to express fds and constraints like *C*, then $\mathrm{FD2}^+$ coincides with the set of fds that are valid in $\mathrm{JOIN}(\overline{R1}, \overline{R2}; X12)$ for any instance $i2 = (R1, R2) \in I2$. We call $\overline{\mathrm{FD2}^+}$ such set of fds.

Theorem 4.3.1 (1) $\overline{\mathrm{FD2}^+} \supseteq \mathrm{FD2}^+$.

(2) *If E holds, then* $\overline{\mathrm{FD2}^+} = \mathrm{FD2}^+$.

The proof appears in Appendix.

The following theorems give sufficient conditions for the conceptual relations defined in Section 2.

Theorem 4.3.2. $A \wedge B \wedge D \Rightarrow S1 < S2$.

Proof. For any $i2=\langle\overline{R1},\overline{R2}\rangle$, let $\bar{f}=\langle f1\rangle$, where $f1(i2)=\text{PROJ}(\text{JOIN}(\overline{R1},\overline{R2});X)$. Given $i1=\langle\bar{R}\rangle\in I1$ let $i2=\langle\overline{R1},\overline{R2}\rangle$, where $\overline{Rj}$ $(j=1,2)$ be any instance of Rj satisfying fds in FD2 such that $\text{PROJ}(\overline{Rj};Xj\cap X)=\text{PROJ}(\bar{R};Xj\cap X)$ (it must exist for hypothesis B).

(1) $i2\in I2$. By hypothesis B.

(2) $i1=\bar{f}(i2)$. In fact:

$$\begin{aligned} f1(i2)&=\text{PROJ}(\text{JOIN}(\overline{R1},\overline{R2});X)\\ &=\text{JOIN}(\text{PROJ}(\overline{R1};X\cap X1),\text{PROJ}(\overline{R2};X\cap X2))\\ &=\text{JOIN}(\text{PROJ}(\bar{R};X1\cap X),\text{PROJ}(\bar{R};X2\cap X))=\bar{R}. \quad\square \end{aligned}$$

Theorem 4.3.3. $A'\wedge B'\wedge C\Rightarrow S2<S1$.

Proof. For any $i1=\langle\bar{R}\rangle$, let $f'=\langle f1',f2'\rangle$, where $fj'(i1)=\text{PROJ}(\bar{R},Xj)$, $j=1,2$. Given $i2=\langle R1,R2\rangle\in I2$, let $i1=\langle\bar{R}\rangle$, where $\bar{R}$ be any instance of R satisfying fds in FD1 such that $\text{PROJ}(\bar{R},X1\cup X2)=\text{JOIN}(\overline{R1},\overline{R2})$.

(1) $i1\in I1$. By hypotheses B' and C.

(2) $i2=\bar{f}'(i1)$. In fact:

$$\begin{aligned} fj'(i1)&=\text{PROJ}(\bar{R};Xj)=\text{PROJ}(\text{PROJ}(\bar{R};X1\cup X2);Xj)\\ &=\text{PROJ}(\text{JOIN}(\overline{R1},\overline{R2});Xj)=\overline{Rj}. \quad\square \end{aligned}$$

Theorem 4.3.4. $A\wedge A'\wedge B\wedge D\Rightarrow S1\sqsubset S2$.

Proof. For any $i1=\langle\bar{R}\rangle$ and $i2=\langle\overline{R1},\overline{R2}\rangle$, let
- $\bar{f}=\langle f1\rangle$ where $f1(i2)=\text{JOIN}(\overline{R1},\overline{R2})$,
- $\bar{f}'=\langle f1',f2'\rangle$ where $fj'(i1)=\text{PROJ}(\bar{R};Xj)$, $j=1,2$.

($\bar{f}$ and $\bar{f}'$ are the same as in Theorems 4.3.1, 2, respectively). Given $i1=\langle\bar{R}\rangle\in I1$, let $i2=\langle\overline{R1},\overline{R2}\rangle$, where $\overline{Rj}=\text{PROJ}(\bar{R}:Xj)$ $j=1,2$.

(1) $i2\in I2$. By hypothesis B.

(2) $i1=\bar{f}(i2)$. See proof of Theorem 4.3.2.

(3) $i2=\bar{f}'(i1)$. Trivial. $\square$

Theorem 4.3.5. $A\wedge A'\wedge B'\wedge C\Rightarrow S2\sqsubset S1$.

Proof. Let $\bar{f}$, $\bar{f}'$ be as in the proof of Theorem 4.3.4. Given $i2=\langle\overline{R1},\overline{R2}\rangle\in I2$, let $i1=\langle\bar{R}\rangle$, where $\bar{R}=\text{JOIN}(\overline{R1},\overline{R2})$.

(1) $i1\in I1$. By hypothesis B'.

(2) $i2=\bar{f}'(i1)$. See proof of Theorem 4.3.3.

(3) $i1=\bar{f}(i2)$. Trivial. $\square$

Corollary 4.3.1. $A\wedge A'\wedge B\wedge B'\wedge C\wedge D\Rightarrow S1\equiv S2$.

All these theorems give sufficient conditions for our conceptual relations. Now we prove that the conditions expressed in Corollary 4.3.1. are also necessary when the query language is the language defined in Section 4.1.

Theorem 4.3.6. $(S1 \sqsubset S2 \wedge C) \Rightarrow (A \wedge A' \wedge B \wedge D)$.

Proof. (1) $S1 \sqsubset S2 \wedge C \Rightarrow A$. If A did not hold, it would exist an attribute Ak such that $Ak \in X-(X1 \cup X2)$. Since $S1 \sqsubset S2$, $\bar{f}$, $\bar{f}' \in Q^+$ should exist such that $\forall i1 \in I1, \exists i2 \in I2$ such that $i1 = \bar{f}(i2)$ and $i2 = \bar{f}'(i1)$. For any $i1$ there would be at least a couple $(Ak: xk)$ in it, which cannot be present in $i2$ because the attribute Ak does not appear in $S2$. So (Fact 4.2.2), there is no $\bar{f} \in Q^+$ such that $i1 = \bar{f}(i2)$.

(2) $S1 \sqsubset S2 \wedge C \Rightarrow A'$. Analogous to Step 1.

(3) $(S1 \sqsubset S2 \wedge C) \Rightarrow B$. Since $\mathrm{FD2}^+ = (\mathrm{FDR1} \cup \mathrm{FDR2})^+$ we have only to prove that $\mathrm{FDR1} \cup \mathrm{FDR2} \subseteq \mathrm{FD1}^+$. We proceed by contradiction. Suppose a fd $Y \to B$ exists (where Y is a set of attributes and B a simple attribute, without loss of generality) such that $Y \to B \notin \mathrm{FD1}^+$ and $Y \to B \in \mathrm{FDR1} \cup \mathrm{FDR2}$. Consider now the instance $i1 = \langle\{t1, t2\}\rangle \in I1$, with $t1.Y = t2.Y$ and $t1.B \neq t2.B$. We will prove that no couple $\bar{f}$, $\bar{f}' \in Q^+$ allows the existence of an instance $i2 \in I2$ such that $i1 = \bar{f}(i2)$ and $i2 = \bar{f}'(i1)$. In fact if $i2 = \langle \overline{R1}, \overline{R2} \rangle \in I2$ satisfies $i2 = \bar{f}'(i1)$, it would be such that

- for every tuple t in $\overline{Rj}$ (where $Xj \supseteq Y \cup \{B\}$) the following holds: $t.Y = t1.Y = t2.Y$ (Fact 4.2.2);
- since Rj satisfies $Y \to B$ and what we said in the former point, $t1.B$ and $t2.B$ cannot be in it at the same time; besides, if $B \in Xk\ j \neq k \in \{1, 2\}$, the value $ti.B$ that is not present in $\overline{Rj}$ cannot be present in $\overline{Rk}$, for the hypothesis C.

If $i2$ satisfies such properties no $\bar{f}$ exists such that $i1 = \bar{f}(i2)$.

(4) $(S1 \sqsubset S2 \wedge C) \Rightarrow D$. We proceed by contradiction. Suppose D is not valid, so we have:

$$(X12 \to X1) \notin \mathrm{FD1}^+ \wedge (X12 \to X2) \notin \mathrm{FD1}^+.$$

Let $B1 \in X1$ and $B2 \in X2$ be two attributes such that $(X12 \to Bj) \notin \mathrm{FD1}^+ (j = 1, 2)$. As a consequence, two instances $i1$, $i1' \in I1$ exist such that:

- $i1 = \langle \bar{R} \rangle$, with $\bar{R} = \{t1, t2\}$;
- $i1' = \langle \bar{R}' \rangle$, with $\bar{R}' = \{t1', t2'\}$ such that:

$$t1.(X-\{B1, B2\}) = t1'.(X-\{B1, B2\})$$
$$= t2.(X-\{B1, B2\}) = t2'.(X-\{B1, B2\}),$$
$$t1.B1 = t1'.B1 \neq t2.B1 = t2'.B1,$$
$$t1.B2 = t2'.B2 \neq t2.B2 = t1'.B2.$$

By the hypothesis $S1 \sqsubset S2$, $\bar{f}, \bar{f}' \in Q^+$ and $i2, i2'$ (with $i2 \neq i2'$ because f' is injective) must exist such that

$$i2 \sqsubset \bar{f}'(i1), \qquad i1 \sqsubset \bar{f}(i2), \qquad i2' \sqsubset \bar{f}'(i1'), \qquad i1' \sqsubset \bar{f}(i2').$$

But $i2$, $i2'$ cannot exist, because each of such instances can contain (by $i2 \sqsubset \bar{f}'(i1)$, $i2' \sqsubset \bar{f}'(i1')$ and Fact 4.2.2) only values of the attributes belonging to $\bar{R}$ and $\bar{R}'$ (that coincide globally) and must contain all such values (by $i1 \sqsubset \bar{f}(i2)$, $i1' \sqsubset f(i2')$ and Fact 4.2.2). The unique instance that satisfies such conditions, with suitable $\bar{f}, \bar{f}' \in Q^+$ is the following:

$$i2 = \langle \overline{R1}, \overline{R2} \rangle \quad \text{where } \overline{R1} = \{t11, t12\}, \overline{R2} = \{t21, t22\}$$

with

$$t11.(X1-\{B1\}) = t12(X1-\{B1\}) = t1(X1-\{B1\}),$$

$$t11.B1 = t1.B1,\ t12.B1 = t2.B1,$$

$$t21(X2-\{B2\}) = t22(X2-\{B2\}) = t2(X2-\{B2\}),$$

$$t21.B2 = t1.B2,\ t22.B2 = t2.B2. \quad \square$$

Consider now the following property (assuming $X = X1 \cup X2$, that is, A and A' hold).

B''. For every $i2 = \langle \overline{R1}, \overline{R2} \rangle \in I2$, $i1 = \langle \text{JOIN}(\overline{R1}, \overline{R2}) \rangle \in I1$.

We will prove with the two following theorems that:

- $A \wedge A' \Rightarrow (B' \Leftrightarrow B'')$,
- $S2 \sqsubset S1 \Rightarrow B'' \wedge C(\wedge A \wedge A')$.

So it will be proved that the condition $B' \wedge C$ is also necessary to have $S2 \sqsubset S1$.

Theorem 4.3.7. $A \wedge A' \Rightarrow (B' \Leftrightarrow B'')$.

Proof. $B' \Rightarrow B''$. The join operation does not destroy any functional dependency. So in JOIN($\overline{R1}, \overline{R2}; X12$) every fd $\in$ FD2$^+$ holds and, by the hypothesis B', every fd $\in$ FD1.

(2) $B'' \Rightarrow B'$. We proceed by contradiction. Suppose that an fd exists such that fd $\in$ FD1$^+$ and fd $\notin$ FD2$^+$. In that case, by Theorem 3.3.1, it should exist at least an instance $i2 = \langle \overline{R1}, \overline{R2} \rangle \in I2$ such that JOIN($\overline{R1}, \overline{R2}$) does not satisfy fd and so JOIN($\overline{R1}; \overline{R2}$) $\notin I1$, against the hypothesis B''. $\square$

Theorem 4.3.8. $S2 \sqsubset S1 \Rightarrow A \wedge A' \wedge B'' \wedge C$.

Proof. (1) $S2 \sqsubset S1 \Rightarrow A$. Analogous to Step 1 of Theorem 4.3.6.

(2) $S2 \sqsubset S1 \Rightarrow A'$. Analogous to Step 2 of Theorem 4.3.6.

(3) $S2 \sqsubset S1 \Rightarrow C$. We proceed by contradiction. Suppose $S2 \sqsubset S1$ and C false. Consider an instance $i2 = \langle \overline{R1}, \overline{R2} \rangle \in I2$, where $\overline{R1} = \{t\}$ and $\overline{R2} = \emptyset$. We will prove that no couple $\bar{f}, \bar{f}' \in Q^+$ allows the existence of an instance $i1 \in I1$ such that $i2 = \bar{f}'(i1)$ and $i1 = \bar{f}(i2)$. In fact, by Fact 4.2.2 the only $i1 \in I1$ such that

$i1=\bar{f}(i2)$ is $i1=\langle\bar{R}\rangle$ with $\bar{R}=\emptyset$, but, by Fact 4.2.1, for every $\bar{f}'\in Q^+$ we have $fi'(i1)=\emptyset$ and so $i2\neq\bar{f}'(i1)$.

(4) $S2\sqsubset S1\Rightarrow B'$. We proceed by contradiction. Suppose an instance $i2=\langle\overline{R1},\overline{R2}\rangle$ exists such that $i2\in I2$ and $\langle\text{JOIN}(\overline{R1},\overline{R2})\rangle\notin I1$. It means that there are a functional dependency $(Y\to B)\in FD1^+$ and two tuples $t1$, $t2\in \text{JOIN}(\overline{R1},\overline{R2})$ such that $t1.Y=t2.Y$ and $t1.B\neq t2.B$. Consider now $i2'=\langle\overline{R1}',\overline{R2}'\rangle$ with $\overline{R1}'=\{t1.X1,t2.X1\}$ and $\overline{R2}'=\{t1.X2,t2.X2\}$; $i2\in I2$ because $\overline{R1}',\subseteq\overline{R1}$ and $\overline{R2}'\subseteq\overline{R2}$ and so $\overline{R1}'$ and $\overline{R2}'$ satisfy all the functional dependencies satisfied by $\overline{R1}$ and $\overline{R2}$. Now we prove that no couple $\bar{f}$, $\bar{f}'\in Q^+$ allows the existence of an instance $i1\in I1$ such that $i2'=\bar{f}'(i1)$ and $i1=\bar{f}(i2')$. In fact if $i1=\bar{f}(i2')$, $i1=\langle\bar{R}\rangle$ should be such that:

- for every tuple $t\in\bar{R}$, $t.Y=t1.Y=t2.Y$,
- only one of the two values $t1.B$, $t2.B$ can be present in $\bar{R}$ (because it satisfies $Y\to B$).

If $i1$ satisfies such properties there exists no $\bar{f}'\in Q^+$ such that $i2'=\bar{f}'(i1)$ (Fact 4.2.2). □

Corollary 4.3.2. $S1\equiv S2\Leftrightarrow(A\wedge A'\wedge B\wedge B'\wedge C\wedge D)$.

Proof. (1) $(A\wedge A'\wedge B\wedge B'\wedge C\wedge D)\Rightarrow S1\equiv S2$. This is Corollary 4.3.1.

(2) $S1\equiv S2\Rightarrow A\wedge A'\wedge B\wedge B'\wedge C\wedge D$.

$$\begin{aligned}S1\equiv S2&\Rightarrow S1\sqsubset S2\wedge S2\sqsubset S1\\&\Rightarrow S1\sqsubset S2\wedge A\wedge A'\wedge B''\wedge C\\&\Rightarrow S1\sqsubset S2\wedge C\wedge A'\wedge B'\wedge A\\&\Rightarrow A\wedge A'\wedge B\wedge B'\wedge C\wedge D.\quad\square\end{aligned}$$

5. Conclusions

In this paper we have characterized various types of conceptual relations between database schemata in the relational model of data. These relations allow to compare database schemata with respect to their ability to represent information and answer to queries.

The examples presented in Section 3 show that our definitions can handle many different practical situations.

Furthermore we have stated necessary and sufficient conditions for our conceptual relations among schemata obtained both by "horizontal" and by "vertical" transformations.

Further research is currently being developed on the extension of our approach to the case in which null values are allowed in the relations.

Appendix, Proof of Theorem 4.3.1.

(1) $FD2^+ \subseteq \overline{FD2}^+$. Since the join operation does not destroy fds in $FDR1^+ \cup FDR2^+$, we have

$$FD2^+ = (FDR1 \cup FDR2)^+$$
$$= (FDR1^+ \cup FDR2^+)^+ \subseteq (\overline{FD2}^+)^+ = \overline{FD2}^+.$$

(2) If E holds, then $FD2^+ = \overline{FD2}^+$.

Let $Y \to B$ be a fd in $\overline{FD2}^+$. We assume in the following that B is a simple attribute without loss of generality. Furthermore, let $B \in X2$ (the proof is analogous for $B \in X1$). If $Y \cup \{B\} \subseteq X1$ the proof is trivial. Otherwise, for any $i2 = \langle \overline{R1}, \overline{R2} \rangle \in I2$ such that $t11 \in \overline{R1}$ and $t21, t22 \subset \overline{R2}$, where

$$t11.X12 = t21.X12 = t22.X12$$

and

$$t21.(Y \cap X2) = t22.(Y \cap X2),$$

by hypothesis $Y \to B \in FD2^+$ we have

$$t21.B = t22.B$$

and so

$$X12 \cup (Y \cap X2) \to B \in FD2^+.$$

We have now three cases.

(1) $X12 \subseteq Y$. In this case $Y \cap X2 \to B \in FD2^+$ and hence $Y \to B \in FD2^+$.

(2) $X12 \not\subseteq Y$, $X2 \cap Y \neq \emptyset$ In this case for any instance $i2 = \langle \overline{R1}, \overline{R2} \rangle \in 12$ such that:

$$\exists t11, t12 \in \overline{R1}, t21, t22 \in \overline{R2}$$

with

$$t11.(X1 \cap Y) = t12(X1 \cap Y),$$

$$t11(X12 - Y) \neq t12.(X12 - Y),$$

$$t21.X12 = t11.X12,$$

$$t22.X12 = t12.X12,$$

$$t21.(X2 \cap Y) = t22.(X2 \cap Y),$$

by hypothesis $Y \to B \in \overline{FD2}^+$ we have

$$t22.B = t21.B$$

and so

$$(X2 \cap Y) \to B \in FD2^+ \quad \text{and} \quad Y \to B \in FD2^+.$$

(3) $X2 \cap Y = \emptyset$. In this case we have $X12 \to B$.

Furthermore, if E holds, we can show that $Y \to X12 \in FD2^+$ and so $Y \to B \in FD2^+$. Suppose by contradiction that $Y \to X12 \notin FD2^+$; then an instance $i2 = \langle \overline{R1}, \overline{R2} \rangle \in I2$ exists such that $t11, t12 \in \overline{R1}$ and $t21, t22 \in \overline{R2}$ with

$$t11.Y = t12.Y$$

$$t11.X12 \neq t12.X12,$$

$$t11.X12 = t21.X12,$$

$$t12.X12 = t22.X12,$$

$$t21.B \neq t22.B.$$

This contradicts the hypothesis $Y \to B \in \overline{FD2}^+$. □

References

[1] P. Atzeni, G. Ausiello, C. Batini and M. Moscarini, Conceptual relations among relational database schemata, Technical Report R. 80-32, Istituto di Automatica, Università di Roma, Italy.

[2] G. Ausiello, C. Batini and M. Moscarini, On the equivalence among database schemata, *Proc. International Conference on Data Bases*, Aberdeen (1980).

[3] G. Ausiello, C. Batini and M. Moscarini, Conceptual relations between databases transformed under join and projection, *Proc. Mathematical Foundations on Computer Science*, Rydzyna (1980).

[4] G. Ausiello, C. Batini and M. Moscarini, Comparisons among conceptual content of databases in the relational model, Technical Report R 79.33, Istituto di Automatica, Università di Roma, Italy.

[5] C. Beeri, P. Bernstein and N. Goodman, A sophisticate's introduction to database normalization theory, *Proc. Conference on Very Large Data Bases*, West Berlin (1978).

[6] C. Beeri, A. Medelzon, Y. Sagiv and J. Ullman, Equivalence of relational database schemes, Technical Report #252, Princeton University, Princeton, NJ (1978).

[7] P. Bernstein, Synthesizing third normal form relations from functional dependencies, *ACM TODS* **1** (4) (1976).

[8] E.F. Codd, A relational model of data for large shared data banks, *Comm. ACM* (June 1970).

[9] E.F. Codd, Further normalizations on data base relational model, *Data Base Systems*, (Prentice-Hall Englewood Cliffs, NJ, 1972).

[10] E.F. Codd, Relational completeness of data base sublanguages, *Data Base Systems* (Prentice-Hall, Englewood Cliffs, NJ, 1972).

[11] R. Fagin, The decomposition versus the synthetic approach to relational database design, *Proc. Conference on Very Large Data Bases*, Tokio (1977).

[12] R. Fagin, Normal forms and relational database operators, *SIGMOD Conference* (1979).

[13] P. Honeyman, R.E. Ladner and M. Yannakakis, Testing the universal instance assumption, *Information Processing Lett.* **10** (1) (1980).

[14] P. Kandzia and H.J. Klein, On the equivalence of databases in connection with normalization, *Proc. International Workshop on Formal Bases for Data Bases*, Toulouse (1979).

[15] J. Rissanen, Independent components of relations, *ACM TODS* **2** (4) (1977).

[16] E. Sciore, Improving semantic specification in a relational database, *SIGMOD Conference* (1979).

[17] J.M. Smith and D.C.M. Smith, Data base abstractions: Aggregation and generalization, *ACM TODS* **2** (2) (1977).