Acta Cybernetica, Tom. 9, Fasc. 4, Szeged, 1990

Preserving two-tuple dependencies under projection

B. Thalheim*   S. Al- Fedhagi†

Abstract

In relational databases, the semantics is modelled by dependencies defined for the whole set of attributes U and the relations on U. Given a set of dependencies on U and a relation R defined on a subset X of U, does there exist an extension of R to a relation R' on U which satisfies the set of dependencies. This problem is analyzed and a general solution is given in the context of two-tuple constraints.

1 Introduction

To achieve a certain level of scientific treatment of its subject-matter, many proposed descriptions of database models adopt different types of symbolic notations. Typically symbols of set theory, formal logic, graph theory, algebra, etc. are utilized to build a semi-formal language to express the main concepts of the subject matter. The relational model is the most obvious example of these models. It is typically described as having "mathematical elegance" and this mathematical characteristic is mentioned as one of its main advantages.

Dependency theory is a sub-field of the theory of relational database [1] that deals with formalizing integrity constraints and studying their mathematical structures. The importance of the theory stems from its implication on the design of the relational database. Dependency theory started with the very known inference rules for functional dependencies called Armstrong axioms and has grown into a very considerable field in the last fifteen years (see for instance [2], [3], [4]).

The influence of dependency theory and normalization theory, in general, on the database design process is definite. The rigorous treatment of the design process followed by the dependency theory is the sole attempt in that direction. Even though its full practicality is still to be proven, dependency theory forms the "showcase" to the claim that the database design field lends itself to formal treatment. Other attempts to formalize the database design process reflect types of "rules-off-thumb" and involve art more than science.

This paper deals with a type of dependency, called propositional dependencies, that are slightly more general than functional dependencies but still weaker than join dependencies. All issues that are related to functional dependencies can be

*Department of Mathematics, Faculty of Science, Kuwait University, KUWAIT

†Department of Electrical & Computer Engineering, Faculty of Engineering and Petroleum, Kuwait University, KUWAIT

analyzed on a better formal ground in the context of propositional dependencies. This is a major motivation for introducing propositional dependencies since there is still a great deal of interest in functional dependencies. Section three introduces few samples of the benefit of injecting propositional dependencies in the relational database issues.

Furthermore, propositional dependencies have their own significance since they form a constraint language. The language is rich as such that it encompasses all two-tuple constraints, still propositional dependencies are simple to identify, understand, and manipulate.

Additionally, propositional dependencies are interesting mathematical objects on their own. The study of their properties and their relationship to several technical issues, e.g., losslessness, may prove to be beneficial in the future.

The primary goal of this work is to show that propositional dependencies can be used to develop better solutions to the problems related to the issue of projected two-tuple constraints. The given set of propositional dependencies are converted in a standard disjunctive normal form and represented as "constraints tableau". The constraints tableaux, can be treated as ordinary relations, hence, different relational operations such as join and project can be applied to these tableaux. Certain modifications to the join and project operations are necessary since the operands are now constraints and not instances. In this paper the following problems are analyzed:

1. Given a set of functional dependencies Σ over the universal set of attributes U, and a relation $S(X), X \subseteq U$ such that S satisfies $\pi_X(\Sigma)$, i.e. the projection of Σ over X, then under what conditions does there exist a relation $R(U)$ that satisfies Σ such that $S = \pi_X(R)$, i.e. S is the projection of R.

2. Since functional dependencies are not preserved under projection, then under what conditions can we have a constraint preserving projection.

These problems are important in the design of the relational database. Let I be the set of all possible relations that satisfies Σ. A desired property of database schemes is that whenever the projection set $\pi_X(I)$ satisfies "projected constraints" it follows that I satisfies the constraints Σ. These problems are connected with three other well known database problems [5].

A. The extensibility problem: Let Σ be a set of integrity constraints on U and let R be a relation which satisfies Σ. Suppose that U' is an extension of U and Σ' is a set of some additional restrictions. Does there exist an extension of R on U' which satisfies both Σ and Σ' ? This problem was considered in the context of weak instances and the realization of the universal relation assumption [6]. Problem 1 gives a partial solution to this problem.

B. The view update problem: Given a conceptual scheme (U, Σ) and a set of views $(U_1, \Sigma_1) \ldots, (U_n, \Sigma_n)$, let $R_1, \ldots, R_n$ be relations for these views. Does there exist an algorithm to decide whether an update of one relation is consistent with the other relations?

C. The implied constraint problem: Let $S_1 = (U_1, \Sigma_2)$ and $S_2 = (U_2, \Sigma_2)$ be two conceptual schemes. Consider the database mapping $\gamma : S_1 \longrightarrow S_2$. Such a mapping induces in a natural way a mapping $\gamma^* : \mathrm{SAT}(S_1) \longrightarrow \mathrm{SAT}\ (S_2)$ where $\mathrm{SAT}(S_i)$ denotes the set of all relations on U_i satisfying Σ_i. We ask whether γ is correct, i.e. $\gamma^*(\mathrm{SAT}(S_1)) = \mathrm{SAT}(S_2)$. In general, this problem is undecidable [7]. A solution is known only for some special cases. Problem 2 can contribute to a general solution of this problem.

Known solutions to problem 1 mentioned previously above require that $S(X)$ should satisfy additional non-two-tuple constraints in order to guarantee the existence of $R(U)$. Our solution characterizes the conditions under which there exists a relation $R(U)$, utilizing only the given two-tuple constraints such as functional dependencies or propositional dependencies. Similarly, we characterize conditions under which we can have a constraint preserving projection, utilizing only two-tuple constraints.

We assume that reader is familiar with relational database theory, and with some background in propositional logic. U is used to denote the set of attributes of the universal relation. X, Y, Z, W (possibly subscripted) are used to denote relation schemes. $S(X), R(X)$ are used to denote relations instances over $X \subseteq U$. The relations $S(X)$ and $R(X)$ may be written as S and R, respectively, when X is understood or immaterial. Small letters u and w are used to denote tuples in relation instances (e.g. $u_1 \in R$). Let us denote by $\underline{R}$ (or $\underline{R}(U)$) the set of all relations on U. The projection of R to a subset X of U is denoted by $R[X]$, i.e. $R[X] = \{u[X] | u \in R\}$ where $u[X]$ is the restriction of u to X.

A relation R satisfies a set of constraints Σ if it satisfies each constraint in Σ. $\mathrm{SAT}(X, \Sigma), X \subseteq U$, is the set of all relations over X that satisfy Σ. $\mathrm{SAT}(X, \Sigma)$ may be written as SAT (Σ) or $\mathrm{SAT}(X)$ when Σ or X are understood respectively.

2 Propositional Dependencies (PDs)

Propositional dependencies form a formal apparatus to express constraints on two-tuples relations. They provide a foundation based on propositional calculus that is suitable for this purpose.

Let $U = \{A_1, \ldots, A_n\}$ be the given set of attributes. With each attribute A there is associated a *propositional variable* A'. For two different tuples t, t' on U, the propositional variable A' denotes the proposition: "The two tuples agree in the A-value". The negation of $A', -A'$, denotes the contrary, that these tuples have different A-values. Without any loss of generality we denote by A the attribute and the propositional variable.

Given the set $\{\wedge, \vee, -, \rightarrow, \leftrightarrow\}$ of logical connectives (conjunction, disjunction, negation, implication, equivalence) and the set U, the set $L(U)$ of *propositions* on U is defined as follows:

1. Any propositional variable is a proposition.

2. If H and H' are propositions then $-H, (H \wedge H'), (H \vee H'), (H \rightarrow H'), (H \leftrightarrow H')$ are propositions.

For any pair of different tuples (t, t') and the set $L(U)$ we define an *interpretation* of propositions as follows:

1. The propositional variable A is true for (t, t'), if $t[A] = t'[A]$ and otherwise A is false.

2. $-H$ is valid for (t, t') if H is false; furthermore, for (t, t'): $H \wedge H')$ is said to be valid for (t, t') if H and H' are valid for (t, t') (if "H and H'"); analogously the validity of $(-H \vee H')$ is defined by "H or H'", $(H \rightarrow H')$ by $(-H \vee H')$ and $(H \leftrightarrow H')$ by $((H' \longrightarrow H) \wedge (H \longrightarrow H'))$.

The validity of H for different t, t' is denoted by $_{(t,t')}| = H$.

For a set of attributes $X = \{B_1, \ldots, B_m)$ the set X is also used to denote the proposition $B_1 \wedge \ldots \wedge B_m$.

The notation $_{(t,t')}| = H$ can be extended to $_R| = H$ as follows:
The proposition H is valid in the relation R (denoted by $_R| = H$) iff for any pair of different tuples (t, t') from R, H is valid; i.e. $_{(t,t')}| = H$.

A set $\underline{H}$ of propositional dependencies is valid in R (denoted by $_R| = \underline{H}$) if all elements of $\underline{H}$ is valid in R. Note that we will use the under bar notation whenever sets of relations or formulae are to be denoted.

For a subset $\underline{R}'$ of $\underline{R}$, a given set $\underline{H}$ of propositional dependencies and a propositional dependency H we say that the set $\underline{H}$ implies H in $\underline{R}$ if for any relation R from $\underline{R}'$ in which $\underline{H}$ is valid, it holds also $_R| = H$ (denoted by $\underline{H}_{\underline{R}'}| = H$ or by $\underline{H}| = H$ for $\underline{R}' \subseteq \underline{R}$).

Corollary 1 *For any relation R with $|R| \leq 1$ and any proposition in $L(U)$; $\underline{H}_R| = H$.*

The "world of two tuple relations" [8] $\underline{R}_2$ denotes the set of two-tuple relations that can be constructed from possible relations with two or more tuples. A two-tuple constraint is a condition that is imposed in the world of two-tuple relations. For example, the proposition $H = ((-X \wedge Y) \vee (X \wedge -Y))$ expresses the following constraint: for any two different tuples t, t', $_{(t,t')}| = H$ iff the two tuples differ in the X-value and match in the Y-value or they match in the X-value and they differ in the Y-value. Functional dependencies are examples of two-tuple constraints.

Any formula from $L(U)$ is called *propositional dependency.*

Corollary 2 *For any set $\underline{R}'$ which contains $\underline{R}_2$, any set $\underline{H}$ of propositional dependencies and a propositional dependency H the following are equivalent:*
(I) $\underline{H}| = H$.
(II) $\underline{H}_{\underline{R}'}| = H$.

Example 1 [1]. Let $U = \{A, B, C, D, E\}, \Sigma = \{A \to E, B \to E, CE \to D\}$. Then the following propositional dependencies are equivalent to the dependencies in Σ:

$-A \vee E$
$-B \vee E$
$-(CE) \vee D$

Σ implies that AC is a key of any relation satisfying Σ. This property is expressed by the propositional dependency $-(AC)$.

Example 2. Suppose that $XY = U$ where XY denotes the union of the sets X and Y. For a functional dependency $X \to Y$, e.g. X is the *key* of U, the equivalent propositional dependency is $-X$. That is, for any two tuples in the relations on U, the two tuples differ in the X-value.

Example 2 shows two propositional formulae that have the same meaning on a given universe U because of the definition of the interpretation of H and the formula $H \wedge -U$. The disjunct $-U = (-A_1 \vee \ldots \vee -A_n)$ for $U = \{A_1, \ldots, A_n\}$ is always assumed because relations are defined to be sets and two tuples of a relation should be different. Therefore, the disjunct $-U$ can be eliminated in all propositional dependencies or can be added to all propositional dependencies. Instead of considering the whole propositional logic $L(U)$ we add to all dependency sets H the axiom $(-A_1 \vee \ldots \vee -A_n)$ as an axiom in our propositional logic called dependency propositional logic, DPL.

Delobel and Casey [9] were the first to relate the functional dependencies to material implications in the two-valued Boolean algebra which is equivalent to propositional logic. In [17], Demetrovics et al. considered the extension of functional dependencies to different classes of Boolean dependencies. Using the theory of Boolean functions, there can be derived different algorithms for scheme design [2]. Propositional dependencies were first introduced by Sagiv et al. as "Boolean dependencies" [10]. They were studied in details, independently by Thalheim [12], Al-Fedaghi [11] and Berman and Blok [13]. In [10] it is claimed that the consequence relation for the class of Boolean dependencies is equivalent to the consequence relation for propositional logic. Unfortunately this is not true because H is a consequence of $(H \wedge -U)$ but $(H \wedge -U)$ is not a consequence of H. We notice that the idea of propositional dependencies is basically a dependency system which can replace the formal system of functional dependencies. There is a set of propositional dependencies that is equivalent to any given set of functional dependencies but not vice versa. For example, the formula $(A \rightarrow (B \vee C))$ is not equivalent to any set of functional dependencies. Therefore the family of propositional dependencies has more expressive power. Furthermore the simplicity of the propositional calculus makes the propositional dependencies a very practical tool.

3 The Propositional Constraints Tabelaux

A formula G is said to be in the disjunctive normal form if G has the form of $G_1 \vee G_2 \vee \ldots \vee G_m, m \geq 1$, where each $G_i, 1 \leq i \leq m$, is a conjunction of literals. A standard disjunctive normal form (SDNF) is a disjunctive normal form where each conjunction contains all propositional variables.

Any set Σ of propositional dependencies corresponds to a unique propositional logic formula in the standard disjunctive normal form. It is sometimes very convenient to work with these standard forms instead of Σ. Several issues such as "equivalence" can be easily analyzed through studying standard forms. A constraint tableau, c-tableau, Σ is a $0-1$ matrix that corresponds to the disjunctive normal form of Σ. This tableau is denoted by $T(\Sigma)$ or T when the set Σ is understood.

Let $\Sigma_N = (C_1 \vee C_2 \vee \ldots \vee C_k)$ be the SDNF of $\Sigma(U)$. Each conjunct C_i includes $m = |U|$ literals. The tableau $T(\Sigma)$ is the $0-1$ matrix where row i corresponds to C_i and column j corresponds to attribute (i.e. propositional variable) $A_j \in U$. The entry (i, j) of T is defined as follows:

$$(i,j) = \begin{cases} 1 & \text{if } A_j \text{ is in } C_i \\ 0 & \text{if } -A_j \text{ is in } C_i \end{cases}$$

Example 3: The set of functional dependencies Σ, given in Example 1 can be represented by the c- tableau of Figure 1.

We use the names of attributes to denote the columns of the c-tableau. Since any row of the c-tableau represents a conjunction of literals in the SDNF, it makes sense to say that the two-tuple relation R satisfies that row. In general, we say that a given relation R satisfies that row or it satisfies a given c-tableau T. Furthermore, SAT(Σ) may be denoted by SAT($T(\Sigma)$) or SAT (T) when Σ is understood.

Definition: Let T be a c-tableau and $\{A_1, \ldots, A_n\}$ be the set of its attributes (i.e. the columns names). Without loss of generality, the projection of T over $X = \{A_1, \ldots, A_m\}, m < n$, is defined as follows:

$$\overline{\pi}_X(T) = \{t | t \quad \text{is the subrow of } T \text{ over the attributes } X \text{ such that}$$

$$(u[A_1] = 0) \vee (u[A_2] = 0) \vee \ldots \vee (u[A_m] = 0)\}$$

That is, the all 1's sub-row is dropped out of $\overline{\pi}_X(T)$.

A	B	C	D	E
0	0	0	0	0
0	0	0	0	1
0	0	0	1	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	0
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1

Figure 1 : $T(\Sigma)$ of dependenciesin example 2

Example 3. (continued). For the tableau presented in Figure 1 the following projections are defined.

A	B	C	D
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	0	1

$\overline{\pi}_{ABCD}(T)$

A	B	C
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0

$\overline{\pi}_{ABC}(T)$

An Ω-tableau $\Omega(X)$ is defined as the two-tuple $< \hat{T}(X), R^2(X) >$ where $\hat{T}(X)$ is a 0-1 matrix and $R^2(X)$ is a set of pairs of labels $\{(u_i, u_j), i < j\}$. An Ω- tableau may correspond to a relation $R(X)$ as follows. The set $R^2(X)$ represents all pairs of tuples of $R(X)$. Hence $|R^2(X)| = (\frac{n}{2}), n = |R(X)|$. A mapping μ is defined between the pairs $\{(u_i, u_j)\}, i \neq j$, in $R^2(X)$ and the rows in $\hat{T}(X)$ as follows:

The A_k-value in row $\mu(u_i, u_j)$ of

$$\hat{T}(X) = \begin{cases} 1 & \text{if } u_i[A_k] = u_j[A_k] \\ 0 & \text{otherwise} \end{cases}$$

Tuple	A_1	A_2
u_1	a_1	b_1
u_2	a_1	b_2
u_3	a_2	b_2
u_4	a_2	b_3

(a) $R(A_1A_2)$

$\hat{T}(A_1 A_2)$	
$A_1\ A_2$	$R^2(A_1\,A_2)$
1 0	u_1, u_2
0 0	u_1, u_3
0 0	u_1, u_4
0 1	u_2, u_3
0 0	u_2, u_4
1 0	u_3, u_4

(b) $\Omega(A_1A_2)$

Figure 2 : Relation and its Ω – tableau.

for all $A_k \in X$.

Example 4: Figure 2 shows a given relation and its corresponding Ω-tableau.

Now we formalize this approach.

For any c-tableau T and a relation R on $U = \{A_1, \ldots, A_n\}$ that satisfies T we can define another extended tableau as follows. Let σ be the following function:

$$\sigma(u, w) = \begin{cases} 1 & \text{if } u = w \\ 0 & \text{if } u \neq w \end{cases}$$

where u, w are values from the domains of R. Then we define a set of rows as follows:

$$\Omega(R) = \{(\sigma_1, \ldots, \sigma_n, b_1, b_2) | \sigma_i = \sigma(b_1[A_i], b_2[A_i])\}$$

where b_1, b_2 are labels of tuples of R.

Notice that for $T(R) = \{(\sigma_1, \ldots, \sigma_n) | (\sigma_1, \ldots \sigma_n, b_1, b_2) \in \Omega(R) \text{ for } b_1, b_2 \in R\}$ and the c-tableau T we get $T(R) \leq T$. Furthermore the projection of $\Omega(R)$ can be formalized as follows:

$\pi_X(\Omega(R)) = \{t | t$ is the projection of a row of $\Omega(R)$ on X leaving the labels out $\}$.

Example 5. Consider the following relations R_1, R_2 defined on $U = \{A, B, C, D\}$ and their Ω-tableaus.

R_1 :	A	B	C	D
u_1	1	3	5	7
u_2	2	4	5	8
u_3	1	4	6	8

$\Omega(R_1)$:	A	B	C	D	t_1	t_2
	0	0	1	0	u_1	u_2
	1	0	0	0	u_1	u_3
	0	1	0	1	u_2	u_3

R_2 :	A	B	C	D
u_1	1	3	5	8
u_2	2	4	5	8
u_3	1	4	6	8

$\Omega(R_2)$:	A	B	C	D	t_1	t_2
	0	0	1	1	u_1	u_2
	0	1	0	1	u_2	u_3
	1	0	0	1	u_1	u_3

We can define arbitrary tableaus in the following way. Given a finite abstract set of tuple names $\{u_1, \ldots, u_m\}$ and a domain set $\{A_1, \ldots, A_n\}$. Then any set

$$\{(\sigma_1, \ldots, \sigma_n, u_i, u_j) | 1 \leq i < j < m, \sigma_k \in \{0, 1, \}\}$$

form an Ω-tableau.

Example 6. Let $u = \{A, B\}$. Consider the following Ω-tableau

Ω	A	B	t_1	t_2
	0	0	u_1	u_2
	1	0	u_1	u_3
	1	0	u_2	u_3

There exists no relation R with $\Omega = \Omega(R)$. To prove this, we show that if R exists then the tuples u_1 and u_2 should be equal on the attribute A. Since $u_1[A] = u_3[A]$ and $u_3[A] = u_2[A]$ then $u_1[A] = u_2[A]$. Clearly, this contradicts the Ω-tableau where $u_1[A] \neq u_2[A]$.

The tableau Ω is said to be realizable if there exists a relation R with $\Omega = \Omega(R)$. Let us first consider the realizability of Ω-tableaus. We define a relation for any attribute $A \in U = \{A_1, \ldots, A_4\}$ as follows:

$\varsigma_{A_i,\Omega}\{(u_1, u_2)|$ (such that there is a row

$$\sigma_1, \sigma_2, \ldots, \sigma_n, u_3, u_4) \in \Omega \quad \text{with} \quad \sigma_i = 1, \quad \text{and}$$

$$((u_3 = u_1, u_4 = u_2) \quad \text{or} \quad (u_3 = u_2, u_4 = u_1)) \quad \text{or} \quad u_1 = u_2\}$$

Theorem 3 *The tableau Ω is realizable iff for all A in U $\varsigma_{A,\Omega}$ is an equivalence relation.*

Proof.

1. For some A in U let $\varsigma_{A,\Omega}$ be not an equivalence relation. Since $\varsigma_{A_i,\Omega}$ is reflexive and symmetric then there exists in Ω three tuples $(\sigma_1^1, \ldots, \sigma_n^1, v_1, v_2)$, $(\sigma_1^2, \ldots, \sigma_n^2, v_2, v_3)(\sigma_1^3, \ldots, \sigma_n^3, v_1, v_3)$ such that $\sigma_i^1 = \sigma_i^2 = 1$ and $\sigma_i^3 = 0$. Consequently, we get $v_1[A_i] = v_2[A_i] = v_3[A_1]$, and $v_1[A_i] = v_2[A_i]$, i.e. a contradiction. Therefore, Ω is not realizable.

2. If for all A in U $\varsigma_{A,\Omega}$ is an equivalence relation and $\{u_1, \ldots, u_m\}$ is the set of abstract tuples used in Ω then we can define a relation $R = \{u_1, \ldots, u_m\}$ using the partitions $P_{A,\Omega}$ defined by $\varsigma_{A,\Omega}$. For $P_{A,\Omega} = \{V_1, \ldots, V_k\}$ where V_i is an equivalence class we define $u_i[A] = j$ iff $v_i \in V_j$. Obviously $\Omega(R)$ is equal to Ω.

For equivalence relations ς_1, ς_2 on $R = \{u_1, \ldots, u_m\}$, the following operations are defined: $\varsigma_1 \wedge \varsigma_2$ (intersection), $\varsigma_1 + \varsigma_2$ (the smallest equivalence relation containing ς_1 and ς_2), and the comparison $\varsigma_1 \leq \varsigma_2(\varsigma_1 \leq \varsigma_2$ iff $(u_1, u_2) \in \varsigma_1$ implies that $(u_1, u_2) \in \varsigma_2)$.

Corollary 4 *Let R be a relation on $U = \{A_1, \ldots, A_n\}$, and $X = \{B_1, \ldots, B_m\}, Y = \{C_1, \ldots C_k\}, Z \subseteq \{D_1, \ldots, D_l\} \subseteq U$. Furthermore assume the Ω-tableau $\Omega(R)$, and the equivalence relations $\varsigma_{A,\Omega(R)}$ for $A \in U$. Then:*

i) $X \to Y$ is valid in R iff $\varsigma_{B_{1,\Omega(R)}} \wedge \ldots \wedge \varsigma_{B_{m,\Omega(R)}} \leq \varsigma_{C_{i,\Omega(R)}}$ for all $i, 1 \leq i \leq k$.

ii) $X \to Y, Z \to Y$ is valid in R iff for all $i, 1 \leq i \leq k$ $(\varsigma_{B_{1,\Omega(R)}} \wedge \ldots \wedge \varsigma_{B_{m,\Omega(R)}}) + \varsigma_{D_{1,\Omega(R)}} \wedge \ldots \wedge \varsigma_{D_{l,\Omega(R)}}) \leq \varsigma_{C_i,\Omega(R)}$

Proof: Suppose that $X \to Y$ is valid in R. Consequently for two tuples u_1, u_2 in R if $u_1[X] = u_2[X]$ then $u_1[Y] = u_2[Y_2]$. Hence, $(u_1, u_2) \in (\varsigma_{B_{1,\Omega(R)}} \wedge \ldots \wedge \varsigma_{B_{m,\Omega(R)}})$ and $(u_1, u_2) \in \varsigma_{C_{i,\Omega(R)}}$ for any $C_i \in Y$.

In $\Omega(R)$ the property $\varsigma_{B_{1,\Omega(R)}} \wedge \ldots \wedge \varsigma_{B_{m,\Omega(R)}} \leq \varsigma_{C,\Omega(R)}$ is easy to check. If for a row $(\sigma_1, \ldots, \sigma_n, u, v)$ in $\Omega(R)$, $\sigma_i = 1$ for all $A_i \in X$ then for any $A_j \in Y$ $\sigma_j = 1$.

Example 7: Consider the relations R_1 and R_2 given in example 5. The relations $R_1, R_2 \in$ SAT $(\{AC \to D, BC \to D\})$. The dependencies $A \to D, B \to$

$D, C \rightarrow D$ are also valid in R_2 whereas only $B \rightarrow D$ is valid in R_1. Furthermore $D \rightarrow B$ is valid in R_1 but $D \rightarrow B$ is invalid in R_2.

1) Let $\Omega = \Omega(R_1)$. The relationships

$$\varsigma_{A,\Omega} \not\leq \varsigma_{B,\Omega}, \varsigma_{A,\Omega} \not\leq \varsigma_{C,\Omega}, \varsigma_{A,\Omega} \not\leq \varsigma_{D,\Omega},$$

$$\varsigma_{B,\Omega} \not\leq \varsigma_{A,\Omega}, \varsigma_{B,\Omega} \not\leq \varsigma_{C,\Omega}, \varsigma_{B,\Omega} \leq \varsigma_{D,\Omega},$$

$$\varsigma_{C,\Omega} \not\leq \varsigma_{A,\Omega}, \varsigma_{C,\Omega} \not\leq \varsigma_{B,\Omega}, \varsigma_{C,\Omega} \not\leq \varsigma_{D,\Omega},$$

$$\varsigma_{D,\Omega} \not\leq \varsigma_{A,\Omega}, \varsigma_{D,\Omega} \leq \varsigma_{B,\Omega}, \varsigma_{D,\Omega} \not\leq \varsigma_{C,\Omega}$$

can be represented by

$\Omega, \leq$	$\varsigma_{A,\Omega}$	$\varsigma_{B,\Omega}$	$\varsigma_{C,\Omega}$	$\varsigma_{D,\Omega}$
$\varsigma_{A,\Omega}$	1	0	0	0
$\varsigma_{B,\Omega}$	0	1	0	1
$\varsigma_{C,\Omega}$	0	0	1	0
$\varsigma_{D,\Omega}$	0	1	0	1

2) Let $\Omega = \Omega(R_2)$. We get the following table

$\Omega, \leq$	$\varsigma_{A,\Omega}$	$\varsigma_{B,\Omega}$	$\varsigma_{C,\Omega}$	$\varsigma_{D,\Omega}$
$\varsigma_{A,\Omega}$	1	0	0	1
$\varsigma_{B,\Omega}$	0	1	0	1
$\varsigma_{C,\Omega}$	0	0	1	1
$\varsigma_{D,\Omega}$	0	0	0	1

Notice that corollary 4 can be extended to propositional dependencies. For instance, the dependency $D \rightarrow A \vee B \vee C$ is valid in R_2. Generally, the dependency $X \rightarrow Y_1 \vee \ldots \vee Y_k$ is valid in R iff for $\Omega = \Omega(R)$

$$\bigcap_{A \in X} \varsigma_{A,\Omega} \leq \cup_{i=1}^{k} \left(\bigcap_{B \in Y_i} \varsigma_{B,\Omega} \right).$$

where $\cup$ denotes the union of sets, i.e. for sets ς_1, ς_2 of sets $\varsigma_1 \cup \varsigma_2 = \{V \mid$ there are $V_1 \in \varsigma_1, V_2 \in \varsigma_2 : V = V_1 \cup V_2\}$.

4 Projections of Constraints and Relations

According to Maier [1] the notion of "projected constraints" is well defined for functional and multivalued dependencies. If $W \subseteq U$, and Σ is a set of functional and multivalued dependencies then $\pi_W(\Sigma)$ consists of those $X \rightarrow Y$ and $X \rightarrow\rightarrow Y$ such that:

i) there is some $X \rightarrow Z$ or $X \rightarrow\rightarrow Z$ in Σ^+ where Σ^+ is the closure of Σ,

ii) $X \subseteq W$, and

iii) $Y = Z \cap W$.

For functional dependencies, it is always assumed that $Y = Z$. Hence, for the given set of attributes U and set of functional dependencies (over U), $\pi_W(\Sigma) = \{X \rightarrow Y$ in $\Sigma^+ XY \subseteq W\}$ where Σ^+ is the closure of Σ.

Example 8. Consider the sets $U = \{A, B, C, D, E\}, \Sigma = \{A \rightarrow E, B \rightarrow E, CE \rightarrow D\}$ of example 1. Then for $X = \{A, B, C, D\}, \pi_x(\Sigma) = \{AC \rightarrow D, BC \rightarrow D\}$.

In a similar way, the projections of disjunctive normal forms and of c-tableaus are defined.
Given a standard disjunctive normal form $\Sigma_N = C_1 \vee C_2 \vee \ldots \vee C_k$ of propositional dependencies over U. If $X \subseteq U$, then the projection of Σ_N over X is defined as the propositional dependency

$$\pi_x(\Sigma_N) = (C_1' \vee C_2' \vee \ldots \vee C_k') \wedge (-A_1 \vee -A_2 \vee \ldots \vee -A_l)$$

where $X = \{A_1, A_2, \ldots A_l\} \subseteq U$ and C_i' is the disjunct produced from C_i after removing all propositional variables not in X.

Example 9. The standard disjunctive normal form of Σ is
$\Sigma_N = -A-B-C-D-E \vee -A-B-C-DE \vee -A-B-CD-E \vee -A-B-CDE$
$\vee -A-BC-D-E \vee -A-BCD-E \vee -A-BCDE$
$\vee -AB-C-DE \vee -AB-CDE$
$\vee -ABCDE \vee A-B-C-DE \vee A-BCDE \vee AB-CDE$
for $X = \{A, B, C, D\}$.
Then $\pi_x(\Sigma_N) = (-A-B-C-D \vee -A-B-CD \vee -A-BC-D$
$\vee -A-BCD \vee -AB-C-D \vee -AB-CD \vee -ABCD$
$\vee A-B-C-D \vee A-BCD \vee AB-CD) \wedge (-A \vee -B \vee -C \vee D)$.

Now let us introduce the extension of Ω-tableaux. Given the sets

$$X = \{A_1, \ldots, A_n\}, Y = \{A_1, \ldots, A_n, B_1, \ldots, B_k\}$$

and a set

$$\Omega = \{(\sigma_1, \ldots, \sigma_n, v, w)\}.$$

A tableau $\Omega' = \{(\sigma_1, \ldots, \sigma_n, \sigma_1, \ldots, \sigma_k, v, w)\}$ defined on Y is said to be an extension of Ω to Y if $\pi_X(\Omega') = \Omega$.
For realizable Ω-tableaux we introduce the set $\Psi_Y = \Psi_Y(\Omega) = \{\Omega' | \Omega'$ is an extension of Ω to $Y.\}$

Example 10. Given $X = \{A, B, C, D\}, Y = \{A, B, C, D, E\}$. The following set of Ω'-tableaux are extensions of $\Omega_1 = \Omega(R_1)$ to Y where R_1 is given in example 5.

Ω'_{11}	A	B	C	D	E	t_1	t_2
	0	0	1	0	0	u_1	u_2
	0	1	0	1	0	u_2	u_3
	1	0	0	0	0	u_1	u_3

Ω'_{12}	A	B	C	D	E	t_1	t_2
	0	0	1	0	1	u_1	u_2
	0	1	0	1	0	u_2	u_3
	1	0	0	0	0	u_1	u_3

Ω'_{13}	A	B	C	D	E	t_1	t_2
	0	0	1	0	0	u_1	u_2
	0	1	0	1	1	u_2	u_3
	1	0	0	0	0	u_1	u_3

Ω'_{14}	A	B	C	D	E	t_1	t_2
	0	0	1	0	0	u_1	u_2
	0	1	0	1	0	u_2	u_3
	1	0	0	0	1	u_1	u_3

Ω'_{15}	A	B	C	D	E	t_1	t_2
	0	0	1	0	1	u_1	u_2
	0	1	0	1	1	u_2	u_3
	1	0	0	0	1	u_1	u_3

Example 11. Given $X = \{A, B, C, D\}, Y = \{A, B, C, D, E\}$ and $\Omega_1 = \Omega(R_2)$ where the relation R_2 is given in example 5. The set $\Psi_Y = \Psi_Y(\Omega_2)$ can be represented by the following table

Ω'_n	$ABCD$	$t_1\,t_2$	$\Omega'_{21}:E$	$\Omega'_{22}:E$	$\Omega'_{23}:E$	$\Omega'_{24}:E$	$\Omega'_{25}:E$
	0 0 1 1	$u_1\,u_2$	0	1	0	0	1
	0 1 0 1	$u_2\,u_3$	0	0	1	0	1
	1 0 0 1	$u_1\,u_3$	0	0	0	1	1

The first problem presented in section 1 can now be stated as follows: Given a set Σ of integrity constraints defined on U and a relation $R \in$ SAT $(\pi_x(\Sigma))$ where R is defined on X, then does there exist a relation $R' \in$ SAT (Σ) such that $R = \pi_x(R')$? This problem is equivalent to the following problem: Given a relation $R \in$ SAT $(\pi_x(\Sigma))$ where R is defined on X and Σ is defined on U. Does there exist a relation $R' \in$ SAT (Σ) such that $\Omega(R') \in \Psi_U(\Omega(R))$?
For a given set Σ of functional dependencies let Eq (Σ) denote the set of relationships defined in corollary 4.

Example 12. Consider Σ and Σ_X given in example 8.

$$\mathrm{Eq}(\{A \to E, B \to E, CE \to D\}) = \{\varsigma_A \leq \varsigma_E, \varsigma_B \leq \varsigma_E, \varsigma_C \wedge \varsigma_E, \leq \varsigma_D\}$$

$$\mathrm{Eq}(\{AC \to D, BC \to D\}) = \{\varsigma_A \wedge \varsigma_C \leq \varsigma_D, \varsigma_B \wedge \varsigma_C, \leq \varsigma_D\}.$$

As a corollary of Theorem 3 and corollary 4 we get directly the solution of the last problem.

Theorem 5 *Let Σ be a set of functional dependencies defined on X, and let X be a subset of U. For a relation R defined on X such that R satisfies $\pi_X(\Sigma)$ there exists an extension R' in SAT (Σ) if and only if there is in $\Psi_U(\Omega(R))$ a set Ω' such that the relationships of Eq(Σ) are fulfilled in Ω'.*

Example 13. Let us continue examples 5, 8, 10, 11, 12.
The relationship $\varsigma_A \leq \varsigma_E$ from Eq (Σ) is violated in $\Omega'_{11}, \Omega'_{12}, \Omega'_{13}.\Omega'_{14}$ violates $\varsigma_B \leq \varsigma_E, \Omega'_{15}$ violates $\varsigma_C \wedge \varsigma_E \leq \varsigma_D$. Therefore there does not exist any relation extending relation $R_1 \in$ SAT $(\pi_x(\Sigma))$ which satisfies Σ. The relationships $\varsigma_A \leq \varsigma_E$ and $\varsigma_B \leq \varsigma_E$ are not valid in $\Omega'_{21}, \Omega'_{22}, \Omega'_{23}, \Omega'_{24}$. The set Eq (Σ) is valid for Ω'_{25}. Therefore there exists a relation R' in SAT (Σ) with $R = \pi_x(R')$. An example of such a relation is the following relation R_3

R_3	A	B	C	D	E
u_1	1	3	5	8	9
u_2	2	4	5	8	9
u_3	1	4	6	8	9

Based on theorem 5, an algorithm can be developed for the computation of an Ω' tableau if there exists such a set for a given R, Σ, and $\Omega(R)$.

Algorithm.

Input. $U = \{A_1, \ldots, A_n\}, X = \{B_1, \ldots, B_k\} \subseteq U, \Sigma$ is a set of functional dependencies defined on U, and the relation R defined on X.

Output. A relation $R' \in$ SAT (Σ) with $R = \pi_X(R')$ if there exists such a relation.

(i) Construct $\pi_X(\Sigma)$.

(ii) Compute $\Omega(R)$.

(iii) Compute Eq (Σ), and Eq $(\pi_X(\Sigma))$

(iv) If $\Omega(R)$ violates Eq $(\pi_X(\Sigma))$ then output that there does not exist a relation R.

(v) Construction of Ψ tables.

1. If for some $Y \subseteq U - X, Z \subseteq X$, there is a dependency $Z \to Y \in \Sigma(Z = \{C_1, \ldots, C_m\}, Y = \{D_1, \ldots, D_p\})$ then copy the 1-entries in the columns of Z to all columns of Y. The result is the table Ω_1.
2. Compute the 1-entries according to theorem 3. (All columns in $U - X$ must be represented by equivalence relations). The result is Ω_2.
3. If for some $Y \subseteq U - X, Z \subseteq X$, there is a dependency $Z \to Y \in \Sigma(Z = \{C_1, \ldots, C_m\}, Y = \{D_1, \ldots, D_p\}$ and a row with 0-entry in one of the Z-columns and for alll Y-columns except one there are 1-entries in that row then write a 0 in the remaining Y-column in that row. The result is Ω_3.
4. If for some column in $U - X, u_i[A] = u_j[A]$ and $u_i[A] = u_k[A]$ then enter 0 in this column for the (u_j, u_k)-row (this is the closure for equivalence relations). The result is Ω_4.
5. If Ω_4 violates $Eq(\Sigma)$ then output that there can not exist such a relation R'.
6. Compute $\Psi(\Omega_4)$ and check against Eq (Σ). If $\Psi(\Omega_4)$ is empty then there is no relation R' satisfying the requirement. If $\Psi(\Omega_4)$ is not empty then use the proof of theorem 3 for the computation of R'.

Example 14 [15]. Given $U = \{A, B, C, E, F, G, H\}, \Sigma = \{A \to G, B \to G, C \to H, E \to H, GH \to F\}, X = \{A, B, C, E, F\}$, and the relation R:

R	A	B	C	E	F
u_1	1	1	1	1	1
u_2	2	2	2	2	2
u_3	3	3	1	2	3
u_4	1	2	3	3	4

We get after (i) in the algorithm:
$\pi_x(\Sigma) = \{AC \to F, AE \to F, BC \to F, BE \to F\}$, and after (ii) the Ω-tableau:

$\Omega(R)$	A	B	C	E	F	t_1	t_2
	0	0	0	0	0	u_1	u_2
	0	0	1	0	0	u_1	u_3
	0	0	0	1	0	u_2	u_3
	0	1	0	0	0	u_2	u_4
	0	0	0	0	0	u_3	u_4
	1	0	0	0	0	u_1	u_4

Since $\Omega(R)$ obeys $\varsigma_A \wedge \varsigma_C \leq \varsigma_F, \varsigma_A \wedge \varsigma_E \leq \varsigma_F, \varsigma_B \wedge \varsigma_C \leq \varsigma_F$, and $\varsigma_B \wedge \varsigma_E \leq \varsigma_F$ we continue with step (v) using Eq $(\Sigma) = \{\varsigma_A \leq \varsigma_G, \varsigma_B \leq \varsigma_G, \varsigma_C \leq \varsigma_H, \varsigma_E \leq \varsigma_H, \varsigma_G \wedge \varsigma_H \leq \varsigma_F\}$. The following table represents the step (v) of the algorithm:

$ABCEF\ t_1t_2$	after applying the first 4 relationships Ω_1GH	equivalence relations on $\Omega_1 : \Omega_2GH$	applying $\varsigma_G \wedge \varsigma_H \leq \varsigma_F$ to Ω_2 : $\Omega_4G\ H$	check of Eq (Σ) in Ω_3
$00000\,u_1u_2$		1 1	1 1	contradiction
$00100\,u_1u_3$	1	1	0 1	
$10000\,u_1u_4$	1	1	1 0	
$00010\,u_2u_3$	1	1	0 1	
$01000\,u_2u_4$	1	1	1 0	
$00000\,u_3u_4$			0 0	

Therefore we conclude that there can not exist a relation R' in SAT (Σ) such that $R = \pi_x(R')$. Using our approach we get usually a set of contradictions or a set $\psi' \subseteq \psi(\Omega(R))$ of candidate Ω-tableaus for extensions of R.

Example 15. Let us continue example 13.
In step (v) we get for $\Omega_1 = \Omega(R_1)$, and $\text{Eq}(\Sigma) = \{\varsigma_A \leq \varsigma_E, \varsigma_B \leq \varsigma_E, \varsigma_C \wedge \varsigma_E \leq \varsigma_D\}$. The following table is the result of the application of the algorithm.

A	B	C	D	t_1	t_2	Ω_{11}	E	Ω_{12}	E	
0	0	1	0	u_1	u_2				1	contradiction
0	1	0	1	u_2	u_3		1		1	
1	0	0	0	u_1	u_2		1		1	

For $\Omega_2 = \Omega(R_2)$ and $Eq(\Sigma)$ we obtain

A	B	C	D	t_1	t_2	Ω_{21}	E	$\Omega_{22} = \Omega_{23} = \Omega_{24}$	E
0	0	1	1	u_1	u_2				1
0	1	0	1	u_2	u_3		1		1
1	0	0	1	u_1	u_4		1		1

In step (vii) we get the relation

R_3	A	B	C	D	E
	1	3	5	8	9
	2	4	5	8	9
	1	4	6	8	9

For the relations and the dependency sets used in examples 13 and 14, Fagin [14] defines the following "curious dependency" using the equality generality dependencies as an additional condition to guarantee the preservation of dependencis under projection:

if $u_1(B) = u_2(B), u_1(C) = u_3(C), u_2(A) = u_3(A)$ and $u_1(D) = u_2(D)$ then $u_1(D) = u_3(D)$.

This condition is not a sufficient condition as it is shown in the following example.

Example 16. Consider $U, \Sigma, X, \pi_x(\Sigma)$ of example 8, and the following relation R:

	A	B	C	D
u_1	1	3	5	7
u_2	2	3	6	8
u_3	2	4	5	9

Obviously, R obeys $AC \rightarrow D$ and $BC \rightarrow D$ and the curious dependency. Nevertheless, there does not exist an extension of R in SAT (Σ).
Using our algorithm we obtain

Ω_4	t_1	t_2	A	B	C	D	E
	u_1	u_2	0	1	0	0	1
	u_1	u_3	0	0	1	0	1
	u_2	u_3	1	0	0	0	1

and $\varsigma_C \wedge \varsigma_E \not\leq \varsigma_D$. This contradicts the dependency $CE \rightarrow D$.
Maier [1] defines the following additional condition: if $u_1[A] = u_3[A], u_2[B] = u_3[B]$, and $u_1[C] = u_2[C]$ then $u_1[D] = u_2[D]$. The relation R_1 used in examples 5, 10, 13 indicates that this condition is not sufficient. It can be shown that for any k there does not exist an equality formula

$$\alpha_1 \wedge \alpha_2 \wedge \ldots \wedge \alpha_k \rightarrow \alpha$$

which could be used as a necessary and sufficient condition for the extensibility of relations in SAT $(\pi_x(\Sigma))$ to relations in SAT (Σ).
Ginsburg and Zaiddan [15] have shown that the projection of Fd-families is not necessarily an FD-family. In example 14 it is possible to show that no Horn formula can be used to express conditions for the extensibility of relations in SAT $(\pi_x(\Sigma))$ to relations in SAT (Σ) (see [8]). For example 14 an equality formula similar to the 'curious' dependency presented in example 15 has the form

$$(\alpha_1 \wedge \alpha_2 \wedge \alpha_3 \wedge \alpha_4) \rightarrow (\beta_1 \vee \beta_2 \vee \ldots \vee \beta_9).$$

Formulae of this form are clearly not Horn formulae.

5 Preserving Constraints

We would like to know the conditions under which whenever the relation $S(X)$ satisfies $\pi_X(T(\Sigma))$ it follows that $S(X)$ is a projection of a relation SAT $(\Sigma(U))$. The relations discussed in examples 14 and 15 show that if Σ is a set of functional dependencies over U then functional dependency families or classes are not preserved under projection.

It should be noted that the above mentioned problem is a special case of the database satisfaction problem. To solve this problem, some definitions are needed.

An all 1's row or subrow over columns X of T will be denoted by $< 1 >_X$. The X may be dropped when it is understood. Similarly, an all 0's subrow over X is denoted by $< 0 >_X$. If a row or subrow over X has at least one zero then it is denoted by $< *0 >_X$.

Let Z^n denotes the $2^n 0 - 1$ strings of length n. Given the c-tableau $T(X)$ then $T_Z(U)$ denotes the c-tableau that is constructed as the Cartesian product

$T(X) \times Z^n$ where $n = |U - X|, X \subseteq U$. For example if $U = ABC, X = AB$, and $T(X)$ is the following c-tableau:

A	B
0	0
0	1

when $n = 1, Z^n = \{0, 1\}$; and $T_Z(U)$ is the following c-tableau:

A	B	C
0	0	0
0	0	1
0	1	0
0	1	1

We define a new table called oc-tableau, denoted as $\mathring{T}(X)$, of a given c-tableau $T(X)$ as follows:

$$\mathring{T}(X) = T(X) \bigcup \{< 1 >_X\}$$

That is, $\mathring{T}(X)$ is constructed from $T(X)$ plus the all 1's row over X.

Definition: The c-join of $T_1(X_1)$ and $T_2(X_2)$, written $T_1 \bar{*} T_2$, is the c-tableau $T(X_1X_2) = \mathring{T}_1(X_1) * \mathring{T}_2(X_2)$.

The c-join operation is defined in terms of the join operation after adding the 1's rows to the c-tableaux participating in the c-join. Notice that $T(X_1X_2)$ in the definition above is a c-tableau and not an oc-tableau, thus the all 1's row is eliminated in $T(X_1X_2)$. By definition the c-tableau does not include an all 0's row.

Example 17. Let $T_1(AB) = \{(0, 1)\}$, and $T_2(BC) = \{(1, 0)\}$, then:

	A	B	C
	0	1	0
$T_1 * T_2 =$	0	0	1
	0	1	0

Let $T_1(AB) = \{(0, 1), (0, 0)\}$ and $T_2(BC) = \{(0, 1), (0, 0)\}$, then:

	A	B	C
	0	1	1
$T_1 * T_2 =$	0	0	1
	0	0	0
	1	0	0

Example 18. Let $\pi_{ABCD}(T)$ be the c-tableau shown in Example 3. As it is discussed in Example 1, $U = ABCDE$ and $U - ABCD = E$ hence, $Z = \{(0), (1)\}$. Figure 4 shows $T_Z(U)$. If $T = T_Z(U)$, we can claim that whenever a relation $R(X)$ satisfies $\overline{\pi}_X(T)$ it follows that $R(X)$ is a projection of a relation SAT (T).

Theorem 6 *SAT* $(\overline{\pi}_X(T(\Sigma))) = \pi_X(SAT(T(\Sigma)))$ *iff* $T(U) = T_Z(T)$.

A	B	C	D	E
0	0	0	0	0
0	0	0	0	1
0	0	0	1	0
0	0	0	1	1
0	0	1	0	0
0	0	1	0	1
0	0	1	1	0
0	0	1	1	1
0	1	0	0	0
0	1	0	0	1
0	1	0	1	0
0	1	0	1	1
0	1	1	1	0
0	1	1	1	1
1	0	0	0	0
1	0	0	0	1
1	0	0	1	0
1	0	0	1	1
1	0	1	1	1
1	0	1	1	1
1	1	0	0	0
1	1	0	0	1
1	1	0	1	0
1	1	0	1	1
1	1	1	1	0

Figure 4 : T_Z of example 18

Proof: Suppose that SAT $\overline{\pi}_X(T(\Sigma))) = \pi_X(\text{SAT}(T(\Sigma)))$. Thus $S_1(X) \in \text{SAT}(\overline{\pi}_X(T(\Sigma)))$ may be joined with any relation $S_2(U-X)$ producing $R = S_1 * S_2$ where $R \in \text{SAT}(T(U))$ since $X \cap (U-X) = 0$. All possible relations over $U - X$ are projections of $R \in \text{SAT}(T)$. Thus, a subrow $t[U-X]$ in $T(U)$ is an element in $Z^n, n = |U-X|$; and $T(U) = T_Z(T(U))$, where $T_Z(T(U))$ is the Cartesian product $\overline{\pi}_X(T(U)) \times Z^n$. Suppose that $T_Z(T(U)) = T(U)$. If $S_1(X)$ satisfies $\overline{\pi}_X(T), X \subseteq U$, then for any relation $S_2(U-X)$, the join $S_1 * S_2$ satisfies $T(U)$. Conversely, if $R(U)$ satisfies $T(U)$ then any projection $\pi_X(R)$ satisfies $\overline{\pi}_X(T)$. Thus $SAT(\overline{\pi}_X(T(\Sigma))) = \pi_X SAT(T(\Sigma)))$.

6 Conclusion

Given a set of propositional dependencies $\Sigma(U)$ and a relation S over $X \subseteq U$, we have identified conditions under which there exists a relation $R \in \text{SAT}(\Sigma(U))$ such that $S = \pi_X(R)$. Also, we have identified the conditions under which whenever the projection $\pi_X(I), X \subseteq U$ satisfies π_X SAT $(T(\Sigma))$ it follows that I satisfies $T(\Sigma)$, where $I =$ SAT $(\Sigma(U))$.

These results are applicable to functional dependencies since they are special type of propositional dependencies. Only propositional dependencies are utilized whereas in [6], [8] and [10] non-two-tuple constraints are suggested. The theoretical significance of our results is clear since our approach is completely new. The practical significance of these results in the area of the relational database is similar to the previously mentioned work.

References

[1] D. Maier: The theory of relational databases. Computer Science Press, Englewood, 1983.

[2] C. Delobel, M. Adiba: Relational databases. North-Holland, Amsterdam, 1985.

[3] J. D. Ullman: Principles of database systems. Computer Science Press, Rockville, 1980.

[4] G. Vossen: Datenbankmodelle, Datenbanksprachen and Datenbank-Management-System. Addison-Wesley, Bonn, 1987.

[5] B. Thalheim: Open problems in Database Theory. Bulletin EATCS, 40, 1989.

[6] P. Atzeni, R. Torlone: Approaches to updates over weak instances. Lecture Notes in Computer Science 364, 1989, 12-23.

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

[8] S.S. Al-Fedaghi, B. Thalheim: Logical foundation for two-tuple constraints in the relational database model. Kuwait 1988. Submitted for publication.

[9] C. Delobel, R.G. Casey: Decomposition of data base and the theory of Boolean switching functions. IBM J. Res. Dev., 17, 5, 1973, 374-386.

[10] Y. Sagiv, C. Delobel, D.S. Parker, R. Fagin: An equivalence between relational database dependencies and a fragment of propositional logic. JACM, 28, 3, 435-453.

[11] S.S. Al-Fedaghi: Dependency theory. Draft copy of this book is available as Technical Report TR 85-0202, Electrical and Computer Engineering Department, Kuwait University, Kuwait 1985.

[12] B. Thalheim: Functional dependencies in relational databases. Journal Inf. Process and Cybernetics, 21,1/2, 1985, 23-33.

[13] J. Berman, W. J. Blok: Positive Boolean dependencies. University of Chicago, Research Report in Computer Sciences, No. 5, June 1985.

[14] R. Fagin: Horn clauses and database dependencies. JACM 29, 4, 1982, 952-985.

[15] S. Ginsburg, S. M. Zaiddan: Properties of functional dependency families. JACM 29,3, 1982, 678-698.

[16] E. Sciore: Improving database schemes by adding attributes. ACM PODS, Atlanta, Georgia, 1983, 379-382.

[17] J. Demetrovics, Gy. Gyepesi: On the functional dependency and some generalizations of it. Acta Cybernetica 5(1981), 295-305.

(Received February 12, 1990)