Characterizations of Multivalued Dependencies and Related Expressions

J.L. Balcázar, J. Baixeries
Departament de LSI
Universitat Politècnica de Catalunya
{balqui, jbaixer}@lsi.upc.es

June 1, 2004

Abstract

We study multivalued dependencies, as well as the propositional formulas whose deduction calculus parallels that of multivalued dependencies, and the variant known as degenerated multivalued dependencies. For each of these sorts of expressions, we provide intrinsic characterizations in purely semantic terms. They naturally generalize similar properties of functional dependencies or Horn clauses.

1 Introduction

Multivalued dependencies (MVD) are a natural generalization of functional dependencies, and an important notion in the design of relational databases. The study of the dependencies in a database schema allows the designer to avoid redundancies in the data, which is useful both to reduce storage needs and to avoid potential anomalies as the database is operated on; furthermore, the query optimizers can profit from this declarative knowledge.

In particular, the presence of functional dependencies that do not result from keys indicates the possibility of decomposing a relation with no information loss. Actually, multivalued dependencies precisely characterize the relations in which a lossless-join decomposition can be performed: the fourth normal form corresponds to relations that cannot be decomposed further through lossless joins, and is characterized by the property that all multivalued dependencies are the result of keys. For a more detailed discussion on multivalued dependencies in this context, the reader can refer also to [6], [7], [8], [19] and [20].

Early works on multivalued dependencies focused on providing a consistent and complete calculus for entailment between dependencies. It so happens that functional dependencies admit a deduction calculus corresponding very closely to Horn clauses: both have a naturally analogous syntax, e.g. $AB \rightarrow C$, where the semantics varies from one case to another: either A, B, and C are propositional variables, and then the expression represents the Horn clause $(\neg A \vee \neg B \vee C)$, or they are attributes in a relation scheme and the expression means that, together, the value of the attributes A and B determines the value of C. However, it is not completely trivial that the very same set of deduction rules is consistent and complete both for logical entailment between Horn clauses and for semantic entailment between functional dependencies. One well-known way of explaining the connection is what we will call the "comparison-based binarization" of a given relation r, namely, a relation derived from r, formed by binary tuples, each obtained from a pair of original tuples from r by attribute-wise comparison. (Processes like this one, together with many other binarizations, are called "scalings" in the field of Formal Concept Analysis [11].) Then it is easy to see that a functional dependency holds in r if and only if the comparison-based binarization, seen as a theory (i.e. a set of propositional models), satisfies the corresponding Horn clause.

Similarly, in [16] a family of propositional formulas is identified, for which a consistent and complete calculus not only exists but, additionally, corresponds to a syntactically identical consistent and complete calculus for multivalued dependencies. We call these multivalued dependency formulas. However, the connection is much less clear than simply considering the comparison-based binarization; and, in particular, the database expressions that most naturally correspond to the propositional formulas under a comparison-based binarization are much more restrictive than plain multivalued dependencies. Such database expressions are called "degenerate multivalued dependencies", and, similarly to functional dependencies, they imply equality of certain attributes in specific tuples under certain conditions, as opposed to multivalued dependencies which imply instead the existence of certain tuples in the relation.

Our present contribution is theoretical in nature. We prove here semantic, intrinsic characterizations of multivalued dependencies, degenerate multivalued dependencies, and multivalued dependency formulas. Our statements present alternative properties that hold for a relation r, or for a propositional theory T, exactly when a given multivalued dependency, respectively degenerated multivalued dependency, holds for r, or when a given multivalued dependency formula holds in T. We believe that they constitute a better understanding of the combinatorial structure of these expressions, and that some of our new facts could lead to advances in some of the areas where these expressions play a major role. Further discussion on this, with a more detailed comparison to existing literature, is deferred to the last section of the paper; but let us advance now that we consider a major candidate area for applying our techniques the discovery (such as in [9]) and analysis (in the sense of concept lattices, [11]) of multivalued dependencies that in fact may hold in an extensional database while possibly not being explicitly declared in the scheme.

2 Multivalued Dependencies

Our definitions and notations from relational database theory are fully standard. We denote $R = \{A_1, \ldots, A_n\}$ the set of attributes, each with a domain $Dom(A_i)$; then a tuple t is a mapping from R into the union of the domains, such that $t[A_i] \in Dom(A_i)$ for all i. Alternatively, tuples can be seen as well as elements of $Dom(A_1) \times \cdots \times Dom(A_n)$. Whenever appropriate, we imagine each tuple having received a unique tuple identifier (tid) so that we use tuples and tid's interchangeably. A relation r over R is a set of tuples. We will use capital letters from the end of the alphabet for sets of attributes, and do not distinguish single attributes from singleton sets of attributes. We denote by XY the union of the sets of attributes X and Y. Our binarization process is simple and quite standard:

Definition 2.1 *For tuples t, t' of a relation r, $ag(t, t')$ (read: "agree") is the set X of attributes on which t and t' have coinciding values: $A \in ag(t, t') \Leftrightarrow t[A] = t'[A]$.*

Sets of attributes naturally correspond to binary tuples, in the standard way: the binary tuple plays the role of the characteristic function of the set. Given a relation r, its comparison-based binarization is formed by all the binary tuples corresponding to the sets $ag(t, t')$ as t and t' run through all the tuples in r.

The following is the standard definition of multivalued dependency. Let X, Y, and Z be disjoint sets of attributes whose union is R. For tuples t and t', denote them as xyz and $xy'z'$ meaning that $t[X] = t'[X] = x$, $t[Y] = y$, and so on. Then:

Definition 2.2 *A* **multivalued dependency** *$X \twoheadrightarrow Y$ holds in r if and only if for each two tuples xyz and $xy'z'$ in r, also $xy'z$ appears.*

As it can be seen, a multivalued dependency $X \twoheadrightarrow Y$ takes into account not only the explicitly mentioned X and Y, but also the rest of the attributes, $Z = R \setminus (X \cup Y)$. In fact, by reversing the roles of xyz and $xy'z'$, we see that the tuple xyz' must appear as well: indeed, it is well-known that, with our notation, $X \twoheadrightarrow Y$ if and only if $X \twoheadrightarrow Z$. In fact, to explicitly record this symmetry

in the notation, we slightly depart from now on of the standard notation and write $X \twoheadrightarrow Y|Z$ for $X \twoheadrightarrow Y$.

We wish a characterization of the dependency in terms of the agree sets of the tuples, or, rather, of the tuple identifiers. Our aim is to know whether the dependency holds by just operating on pairs of tids associated to each agree set. Our characterization mainly rests on the following definition:

Definition 2.3 *For a set of attributes X of R, $\tau_r(X)$ is the set of all pairs of tuples (or tid's) from r whose agree set is X:*

$$\tau_r(X) = \{\langle t, t'\rangle | ag(t, t') = X\}$$

From now on, we consider r fixed and therefore we drop the subscript. Note that a symmetry property holds, namely, $\langle t, t'\rangle \in \tau(X) \Leftrightarrow \langle t', t\rangle \in \tau(X)$, and that in that case t and t' must differ in all the attributes not in X. We are after some form of algebraic test on the τ values of sets of attributes that characterizes those relations r where a given dependency holds. Consider the following operation:

Definition 2.4 *For sets T, T' of pairs of tuples, we denote $T \bowtie\!\bowtie T'$ the set*

$$T \bowtie\!\bowtie T' = \{\langle t, t'\rangle | \exists t''(\langle t, t''\rangle \in T \wedge \langle t'', t'\rangle \in T') \wedge \exists t'''(\langle t', t'''\rangle \in T \wedge \langle t, t'''\rangle \in T')\}$$

Like a join, this operation is a composition of relations, with some peculiarities such that the composed relations being symmetric and of arity two. In fact, the second condition about t''' is there just to ensure that the outcome is still symmetric. Our main result about mutivalued dependencies is as follows:

Theorem 2.5 *Let X, Y, Z be pairwise disjoint sets of attributes of R, such that their union XYZ includes all the attributes. Then the multivalued dependency $X \twoheadrightarrow Y|Z$ holds in r if and only if, for each $X' \supseteq X$, $\tau(X') = \tau(X'Y') \bowtie\!\bowtie \tau(X'Z')$, where $Y' = Y \setminus X'$ and likewise $Z' = Z \setminus X'$.*

This theorem follows from a sequence of lemmas that we develop in the rest of this section.

Lemma 2.6 $\tau(XY) \bowtie\!\bowtie \tau(XZ) \subseteq \tau(X)$.

(This fact holds irrespectively of the multivalued dependency.)
Proof. Assume that $\langle t, t'\rangle \in \tau(XY) \bowtie\!\bowtie \tau(XZ)$; thus there is a tuple t'' such that $\langle t, t''\rangle \in \tau(XY)$, so that they in particular agree in X, and $\langle t'', t'\rangle \in \tau(XZ)$ so that they also agree in X. (We do not need the second half of the definition of $\bowtie\!\bowtie$ here.) Hence, t and t' agree in X as well. To prove that $ag(t, t') = X$, we see that they differ in all the attributes not in X, that is, either in Y or in Z. Let $B \in Z$: t and t'' disagree in B, or otherwise they would not be together in $\tau(XY)$ but rather would have B inside their agree set. Now, t'' and t' do coincide in B, so that t and t' disagree in B. A symmetric argument shows that they disagree in Y, and thus $\langle t, t'\rangle \in \tau(X)$. Note that we are using here the assumption that X, Y, and Z are disjoint. ∎

Lemma 2.7 *If $X \twoheadrightarrow Y|Z$ holds in R, then $\tau(X) = \tau(XY) \bowtie\!\bowtie \tau(XZ)$.*

Proof. By the previous lemma, we only need to prove the left to right inclusion. Let $\langle t, t'\rangle \in \tau(X)$, so that they do coincide in X. Then the multivalued dependency implies that there is a tuple t'' such that $t[X] = t'[X] = t''[X]$, with $t''[Y] = t[Y]$, $t''[Z] = t'[Z]$, so that $\langle t, t''\rangle \in \tau(XY)$ and $\langle t'', t'\rangle \in \tau(XZ)$; and likewise for the second half of the definition of $\bowtie\!\bowtie$. Thus, $\langle t, t'\rangle \in \tau(XY) \bowtie\!\bowtie \tau(XZ)$. ∎

Lemma 2.8 *If $X \twoheadrightarrow Y|Z$ holds in R, and $X' \supseteq X$, then $\tau(X') = \tau(X'Y') \bowtie\!\bowtie \tau(X'Z')$, where $Y' = Y \setminus X'$ and likewise $Z' = Z \setminus X'$.*

Proof. By the augmentation rule for multivalued dependencies, if $X \twoheadrightarrow Y|Z$ holds in R then $X' \twoheadrightarrow Y'|Z'$ holds as well; then, by the previous lemma, the required equality holds. ∎

Lemma 2.9 *If, for each $X' \supseteq X$, $\tau(X') = \tau(X'Y') \bowtie \tau(X'Z')$, where $Y' = Y \setminus X'$ and likewise $Z' = Z \setminus X'$, then $X \twoheadrightarrow Y|Z$ holds in R.*

Proof. First note that $X'Y' = XY$, and that $X'Z' = XZ$; this is easy to check by simple set-theoretic arguments. Let $\langle t, t' \rangle$ coincide in X; this means that $ag(t, t') = X' \supseteq X$, and therefore $\langle t, t' \rangle \in \tau(X') = \tau(X'Y') \bowtie \tau(X'Z')$. Thus, there is a tuple t'' such that $\langle t, t'' \rangle \in \tau(X'Y')$ and $\langle t'', t' \rangle \in \tau(X'Z')$, so that $t[X] = t'[X] = t''[X]$, with $t''[XY] = t[XY]$, $t''[XZ] = t'[XZ]$; and similarly for the symmetric case, using the second half of the definition of $\bowtie$, so that indeed the multivalued dependency $X \twoheadrightarrow Y|Z$ holds. ∎

Taken together, these lemmas prove our main theorem of this section.

3 Multivalued Dependency Clauses

In this section we will work only with binary tuples, so that the attributes now play the role of propositional variables, and each binary tuple can be seen as a propositional model. Literals, terms, and propositional formulas are defined in the usual way, and are satisfied by a model x if they evaluate to true on it; we use the standard notation $x \models F$. Sometimes we say that a model violates a formula to mean that it evaluates to false on the model. The ordering between models is the boolean-cube bitwise partial order, denoted $x \leq y$, or $x < y$ for the proper order. Operations $\wedge$ and $\vee$ on models apply bitwise. We denote $\top$ the model consisting of all trues. Bitwise unions and intersections are extended to theories (that is, sets of models) in the usual way; we also agree to the standard convention that the union of an empty theory is the all-false model $\bot$, whereas the intersection of an empty theory is the top model $\top$. We frequently overload the numeral notation by denoting true as 1 and false as 0.

The Hamming weight of a model is the number of variables it assigns to true; the Hamming weight of a set of models is the sum of the Hamming weights of its elements.

Horn clauses are disjunctions where either no positive literal appears, or exactly one positive literal appears; the latter are called definite Horn clauses.

The following characterization is known since the earliest works on Horn logic, and holds also beyond propositional domains:

Theorem 3.1 *1. A propositional theory is a Horn theory, that is, can be axiomatized by a conjunction of Horn clauses, if and only if it is closed under intersection.*

2. The smallest Horn theory that contains a given theory is its closure under intersection.

For this propositional case, the proof is not difficult and can be found in a number of references (for instance in [13]). For the sake of comparison with our later contribution, we however state the following easy but crucial step in the proof:

Lemma 3.2 *Consider a Horn clause, and two models x and y that satisfy it. Then $x \wedge y$ also satisfies it.*

From this lemma it immediatelly follows that each Horn clause satisfied by a theory T is also satisfied by the closure of T under intersection.

As mentioned in the introduction, the calculus for entailment in functional dependencies mirrors a calculus for Horn clauses, in the sense that it is easy to associate a Horn clause to each functional dependency in such a way that this mapping commutes with the logical consequence relation; moreover the rules of the calculus are syntactically equal in both sides. Similarly, it turns out that there is a calculus for multivalued dependencies that mirrors, in the same sense of commuting with the logical consequence relation, a calculus for a specific family of propositional formulas: multivalued dependency formulas. They are defined as conjunctions of clauses of the form $X \rightarrow Y \vee Z$, for disjoint terms X, Y, and Z that satisfy the additional condition that

their union is R, the set of all the variables. These implications are naturally called multivalued dependency clauses. See [16] for details on all these issues.

Our main result in this section is a characterization, in the spirit of the closure under intersection of Horn theories, for theories defined by multivalued dependency formulas. Our main technical ingredient is as follows.

Definition 3.3 *Consider a set of binary tuples T. We say that $x \in T$ is a focus of T if, for every $y \in T$, $x \vee y$ is not the top model $\top$.*

Note that, for $y = x$, this implies that the focus x itself is not $\top$. Note also that there may be multiple foci of a given T. In the particular case of a set of two models, either both or neither are foci. Additionally, a theory containing $\top$ has no foci at all.

Definition 3.4 *Consider a propositional theory T. A focused intersection of T is a model that can be obtained as the intersection of all the members of a subtheory $T' \subseteq T$ that has at least one focus (of T').*

We are ready for the main result of this section: a semantic characterization of the theories defined by Horn or multivalued dependency clauses.

Theorem 3.5 *A propositional theory can be axiomatized by a conjunction of Horn or multivalued dependency clauses if and only if it is closed under focused intersection.*

Again we prove this statement through a series of lemmas. The first one is the analogue of lemma 3.2.

Lemma 3.6 *Consider a multivalued dependency clause, and a propositional theory T that satisfies it. Assume that T has a focus. Then the intersection of all the members of T satisfies the clause.*

Proof. Let the clause be $X \to Y \vee Z$. If there is a model $y \in T$ with $y \not\models X$ then the intersection also has the same property, and therefore satisfies the clause. Thus, we assume that all elements of T, including some fixed focus x of T, satisfy X, and by (the soundness of) modus ponens they satisfy either Y or Z; we classify them into T_Y and T_Z accordingly, and remove the focus x from whichever side it fell into.

It is easy to see that the intersection y_Y of all the models in T_Y (even if this set is empty) satisfies Y. A bit less obviously, for all $z \in T_Z$, the zeros of z are in Y, since z satisfies XZ, and then $XYZ = R$ implies that the complement of XZ is included in Y; so that the intersection y_Z of T_Z also has all the zeros in Y. Observe that the intersection of all of T, by associativity and commutativity, is exactly $x' = x \wedge y_Y \wedge y_Z$.

Recall that x satisfies X and, by the implication, also either Y or Z. Assume first it satisfies XZ: since $y_Y \models Y$, and XYZ has all the variables, $x \vee y_Y = \top$, and $x \vee y = \top$ as well for every $y \in T_Y$ because $y_Y \leq y$: then x would not be a focus. The only way is that $T_Y = \emptyset$ so that no such y exists. Then $y_Y = \top$, and x' satisfies Z since it is the intersection of three models that do.

The other case, dually, is when $x \models XY$. As we just saw, y_Z has all the zeros in y, so $x \vee y_Z = \top$, and we argue exactly as in the previous case. ∎

In general, we will apply this lemma to subtheories, as in definition 4. Now we can concentrate on the forward direction of our main characterization.

Lemma 3.7 *Assume that T is axiomatized by a conjunction of Horn or multivalued dependency clauses. Then T is closed under focused intersection.*

Proof. Let $T' \subseteq T$ have a focus x. Consider each of the clauses that participate in the axiomatization: whether they are Horn, or multivalued dependency clauses, by the previous lemmas we know that they are satisfied by the intersection x' of all the models in T'; therefore, x' satisfies all the axioms of T, whence it must belong to T. ∎

Let us prove the converse now. Assuming that T is closed under focused intersection, we must show how to axiomatize it by a conjunction of Horn or multivalued dependency clauses: we simply consider all the clauses of these sorts that are true for all of T. We must prove that their conjunction axiomatizes T, that is, a model x satisfies them all if and only if $x \in T$. The "if" part is obvious since these clauses are all true for all of T. Thus it remains to see that a model x that is not in T violates some such clause that is true of T. Our notation will rely strongly on the following definition: for a theory T and a model x, the subset $T_x \subseteq T$ is

$$T_x = \{y \in T \mid x \leq y\}$$

The following is argued essentially as in the characterization of Horn theories, but we sketch the proof for the sake of completeness. It covers the case where the intersection of the models in T above x remains above x.

Lemma 3.8 *Let $x \notin T$. Consider the intersection z of all the models in T_x. If $x \neq z$, then x violates a Horn clause that is true of T.*

Proof. Clearly $x \leq z$, so that if they differ then $x < z$. Let X be the variables satisfied by x. We consider a clause of the form $\phi = (X \rightarrow v)$ where $v \in V$ is a variable that is true in z and false in x; clearly $x \not\models \phi$. However, $T \models \phi$: indeed, for each $y \in T$, either $y \notin T_x$, and then it falsifies $1(x)$, or $y \in T_x$, in which case $z \leq y$ forces v to be true in y so that $y \models \phi$ as well. ∎

The final case corresponds to x being indeed the intersection of all of T_x, which no longer means that it must be in T since the theory may not be closed under arbitrary intersections: we are actually assuming that $x \notin T$. Let us discard beforehand a special case, namely, $x = \top$, or $T_x = \emptyset$. This is easily handled by the nondefinite Horn clause $\bigwedge_{v \in V} v \rightarrow \Box$: it excludes $\top$ and no other model, so that it is indeed satisfied by T (since $x = \top \notin T$) and falsified by x.

Thus from now on we assume that $x < \top$. We prove that this case is covered by the multivalued dependency clauses.

Lemma 3.9 *Let T be a theory closed under focused intersection, and let $x \notin T$, with $x < \top$. Assume that the intersection of all the models in T_x is precisely x. Then x violates a multivalued dependency clause that is true of T.*

Proof. Consider a subset $T' \subseteq T_x \subseteq T$ such that x is still the intersection of all the models in T', but T' has, under this condition, minimal Hamming weight. Note that, in particular, this implies that each pair y, z in T' reaches $y \vee z = \top$; otherwise, $y \wedge z$ would belong to T by closure under focused intersection, thus to T_x as well, and replacing both in T' by this intersection would reduce the Hamming weight.

As a consequence, fixed any $y \in T'$, all the other elements of T', and their intersection as well, have value true for all those variables that y sets to false. Note also that x is not all true and thus T' is nonempty.

Pick any arbitrary $y \in T'$; since $x \notin T$ but $y \in T_x$, they differ, and $x < y$. Let z_y be the intersection of $T' - \{y\}$, so that $x = y \wedge z_y$. As just argued in the previous paragraph, $y \vee z_y = \top$. Also, $T' - \{y\} \neq \emptyset$ since otherwise $x = y$ (but note that z_y may not be in T).

Let X be the variables satisfied by x, and likewise Y and Z for y and z_y respectively. Consider the clause $\phi = (X \rightarrow Y \vee Z)$, which is then a multivalued dependency clause (technically, the ones of x should be removed from both disjuncts of the right hand side but this is in fact irrelevant). The minimality of T' (and the fact that $x < y$, so y is not all zeros) implies that $x < z_y$ since otherwise we could cross y off from T' and reduce Hamming weight.

Therefore, $x < y$ and $x < z_y$, which jointly imply that x falsifies ϕ. We prove now that in fact T satisfies it, so that it belongs to the axiomatization we constructed in the first place, and this completes the proof that each model not in T falsifies at least one of the axioms, which is our current claim.

Assume, therefore, that some model $w \in T$ falsifies this clause; that is, it satisfies its left hand side but falsifies both disjuncts of the right hand side. Satisfying the term X means $x \leq w$, so

that $w \in T_x$. Falsifying Y implies that $w \wedge y < y$. If we can prove that $w \wedge y \in T$ then we are done, since both are above x, thus both are in T_x and $w \wedge y \in T_x$ as well: it could have been used instead of y in T', contradicting again the minimality of T'.

Here is where closure under focused intersection plays its role: we simply prove that $w \vee y < \top$, and since both w and y are in T, their focused intersection must be as well. Thus it only remains to prove that w and y have a common zero, and for this we use the single remaining property of w, that of not satisfying the second disjunct of the right hand side of ϕ. Namely, $w \not\models Z$ means that Z intersects the zeros of w. Now, recalling $x \leq w$, the zeros of w, say $0(w)$, must be also zeros of x, say $0(x)$ with $0(w) \subseteq 0(x)$; hence $0(w) = 0(w) \cap 0(x)$, and $x = y \wedge z_y$ so that $0(x) = 0(y) \cup 0(z_y)$ for likewise defined $0(y)$ (the complement of Y) and $0(z_y)$ (the complement of Z):

$$Z \cap 0(w) = Z \cap 0(w) \cap 0(x) = Z \cap 0(w) \cap (0(y) \cup 0(z_y))$$

Applying distributivity,

$$Z \cap 0(w) = (Z \cap 0(w) \cap 0(y)) \cup (Z \cap 0(w) \cap 0(z_y))$$

where the second argument of the union is obviously empty; therefore, given that $Z \cap 0(w) \neq \emptyset$ the set $Z \cap 0(w) \cap 0(y)$ is equally nonempty and the larger set $0(w) \cap 0(y)$ is nonempty too, as was to be shown. ∎

Taken together, the lemmas prove the main theorem in this section. We can use the same techniques to characterize the case of using only multivalued dependency clauses, without the company of Horn clauses; the property is somewhat less elegant. We first note that a model with less than two variables set to false satisfies every multivalued dependency clause.

Theorem 3.10 *A propositional theory T can be axiomatized by a conjunction of multivalued dependency clauses if and only if it is closed under focused intersection, contains $\top$ and, for each model $x \notin T$, $x = \bigwedge T_x$.*

We only sketch the proof since it uses the same techniques: the "if" part follows from lemma 3.9, by considering all the multivalued dependency clauses satisfied by T, and picking $x \notin T$ (so $x \neq \top$); then the properties about x and the closure of T are exactly what we need to apply lemma 3.9 and prove that x falsifies some of these clauses; thus the conjunction of these classes axiomatizes T. For the "only if" part, the observation just before this theorem says that T contains $\top$, and lemma 3.7 proves that it is closed under focused intersection; pick $x \notin T$, and consider T_x. To prove $x = \bigwedge T_x$ it suffices to see that, for each $v \in 0(x)$, there is $y \in T_x$ for which $v \in 0(y)$: indeed, such y is the model that sets only v to zero. Then, since in this case $x \leq y$ and $y \in T$, we have that $y \in T_x$.

4 Degenerated Multivalued Dependencies

We resume now the notational context as in the section on multivalued dependencies. Motivated by the multivalued dependency clauses that we have studied in the previous section, we now consider expressions on a relation r that correspond exactly to imposing a multivalued dependency clause on the comparison-based binarization of r.

Indeed, this means that we consider pairs of tuples from r, of the form $\langle t, t' \rangle$, and, on the basis of a multivalued dependency clause $X \rightarrow Y \vee Z$, we require that, if $t[X] = t'[X]$, then either $t[Y] = t'[Y]$ or $t[Z] = t'[Z]$: this is equivalent to requiring that the clause $X \rightarrow Y \vee Z$ holds for the comparison-based binarization of r.

As in previous references [16], we use a double arrow for the degenerated dependency:

Definition 4.1 *A degenerated multivalued dependency (DMVD) $X \twoheadrightarrow Y|Z$ holds in a relation if for each pair of tuples t, t' such that $t[X] = t'[X]$ then $t[Y] = t'[Y]$ or $t[Z] = t'[Z]$.*

One simple way of characterizing them is as follows. Consider the following more relaxed form of comparing two tuples on some attributes: $\rho(X) = \{\langle t_1, t_2\rangle | t_1[X] = t_2[X]\}$. Its difference with τ is, clearly, that it is not necessary that those two tuples disagree in the rest of the attributes. The following relationship trivially holds:

Proposition 4.2 $\rho(X) = \bigcup_{X \subseteq X'} \tau(X')$

With that notation, we can easily see the following characterization:

Proposition 4.3 *A degenerated multivalued dependency $X \twoheadrightarrow Y|Z$ holds if and only if $\rho(XY) \cup \rho(XZ) = \rho(X)$.*

Proof. $\Leftarrow$/ Each pair of tuples that agree in X appear in $\rho(X)$, and because of the equality, they also appear in $\rho(XY)$ or in $\rho(XZ)$, which means that they also agree in Y or in Z. Thus, $X \twoheadrightarrow Y|Z$ holds.

$\Rightarrow$/ Pick a pair in $\rho(X)$. If $X \twoheadrightarrow Y|Z$ holds, it means that for each such pair of tuples t_1, t_2, if $t_1[X] = t_2[X]$ then, the following must hold: $t_1[Y] = t_2[Y]$ or $t_1[Z] = t_2[Z]$. If the former happens, the pair will belong to $\rho(XY)$, otherwise, it will belong to $\rho(XZ)$. Hence, $\rho(XY) \cup \rho(XZ) \supseteq \rho(X)$. The converse inclusion is obvious. ∎

A bit less trivial is the fact that we can also use τ to characterize these formulas, using essentially the same argumentation from the perspective of the larger sets of attributes X':

Proposition 4.4 *$X \twoheadrightarrow Y|Z$ holds if and only if, for each $X' \supseteq X$ with $\tau(X') \neq \emptyset$, either $XY \subseteq X'$ or $XZ \subseteq X'$.*

That is, equivalently, for any proper subsets $Y' \subset Y$ and $Z' \subset Z$, $\tau(XY'Z')$ must be empty.

Proof. $\Rightarrow$/ Let $X \subseteq X'$ with $\tau(X') \neq \emptyset$, and let $\langle t, t'\rangle \in \tau(X')$: then $t[X] = t'[X]$, and by the fact that the dependency holds we obtain that either $t[Y] = t'[Y]$ or $t[Z] = t'[Z]$; that is, either $Y \subseteq ag(t, t') = X'$ or $Z \subseteq ag(t, t') = X'$.

$\Leftarrow$/ We prove that the dependency holds: assume that $t[X] = t'[X]$, and let $X' = ag(t, t') \supseteq X$; also, clearly $\langle t, t'\rangle \in \tau(X')$, which is thus nonempty. Then it must be that either $XY \subseteq X' = ag(t, t')$, so that they coincide on Y, or $XZ \subseteq X' = ag(t, t')$ so that they coincide on Z. ∎

5 Discussion

We have described a semantic characterization of multivalued dependencies in relational databases, as well as a semantic characterization of the propositional theories axiomatized by conjunctions of Horn clauses and multivalued-dependency clauses, that are their counterpart in the realm of propositional logic. Specifically, we have identified a form of closure under intersection that holds exactly for these theories. This can be seen as analogous to the characterization of Horn theories, that are the parallel in the propositional realm to functional dependencies, as exactly the theories that are closed under unrestricted intersection.

Our interest in these properties stems from recent studies of related data mining problems. Whereas dependencies and other integrity constraints are expected to be identified at the time of designing a database schema, it may actually happen that some such correlation went undetected in the design of the database, and, therefore, it may be possible that actually the relation can be decomposed, i.e. a certain multivalued dependency actually holds on the relation, but is not explicitly documented in the intensional database. One may wish to explore the possibility that some implicit dependency actually holds on the extensional database, that is, the tuples themselves, in order to improve the design, efficiency, and understanding of the phenomena that the database is intended to reflect. In fact, then, as argued in [9], the problem becomes one of inductive analysis, in the standard machine-learning setting of learning from examples. Yet another process that falls in the same analogy is the search for deterministic association rules [1], [15].

Under the name of Discrete Deterministic Data Mining, the proposal has been put forth of computing, from relational data, so-called deterministic association rules, which are association rules with no condition on the support but 100% confidence; and it has been shown that, particularly in scientific domains amenable to automated scientific discovery processes, where correlations between observations are ubiquitous since they are due to underlying natural laws, this sort of data mining process is highly effective [15]. The process can be explained in terms of Concept Lattices since the rules obtained have a precise meaning in the context of Formal Concept Analysis. We contributed to that study [4] by proving that, from a point of view that can be seen as Knowledge Compilation [18], the process of discovery of deterministic association rules is actually constructing (an axiomatization of) the empirical Horn approximation: the smallest Horn theory that contains the given tuples. The translation of these facts into functional dependencies through the comparison-based binarization is quite simple, see [11].

Several algorithms have been proposed in the literature to find functional dependencies from the extensional database; we should mention here [5], [14], [12], [9] and [17]. In particular, TANE [12] is based on partitions of the set of tuples. A recent work [3] has characterized this approach as well in terms of Formal Concept Analysis, and, by combining it with the so-called dependency basis, this partition-based connection with Concept Lattices has been extended to incorporate multivalued dependencies into the framework [2]. What was missing to complete the picture, though, was a connection of multivalued dependencies with concept lattices through comparison-based binarizations instead of using sets of partitions. The results in the present paper started as an attempt to complete this view. However, the need to consider tid pair lists (the operator τ) makes it difficult (or artificial) to describe multivalued dependencies as concept lattices.

According to [9], their algorithm fdep is more efficient than TANE in many empirical evaluations of the computation of functional dependencies. Besides, an important property of the approach of [9] is that, by encapsulating into some subroutines the test of whether a dependency holds for a database, the same algorithmic schemas can be applied to the discovery of multivalued dependencies [10]. Our characterizations have a monotonicity pattern (through the "for all $X' \supseteq X$" condition) that makes them attractive to consider their potential combination with existing algorithms for the discovery of multivalued dependencies, like mdep [10].

More generally, we believe that further advances may be possible through this fundamental study, based on formally proving intrinsic combinatorial characterizations of multivalued dependencies and related expressions. These may suggest either alternative algorithmic avenues, or improvements on existing algorithms by means of, e.g., more aggressive pruning of the search spaces. They may yield views of a higher abstraction level, which could provide further unifying approaches. Further research along these lines is under development.

References

[1] Agrawal R., Mannila H., Srikant R., Toivonen H., Verkamo I. *Fast Discovery of Association Rules.* Advances in Knowledge Discovery and Data Mining, p. 307-328. AAAI Press, 1996.

[2] Baixeries J. *Using Concept Lattices to Model Multivalued Dependencies.* Submitted: http://www.lsi.upc.es/∼jbaixer/recerca/index_recerca.html.

[3] Baixeries J. *A Formal Concept Analysis Framework to Model Functional Dependencies.* To be presented in Mathematical Methods for Learning (2004).

[4] Balcázar, J.L., Baixeries J. *Discrete Deterministic Data Mining as Knowledge Compilation.* Workshop on Discrete Mathematics and Data Mining in SIAM International Conference on Data Mining (2003).

[5] Castellanos M., Saltor F. *Extraction of Data Dependencies.* Information Modelling and Knowledge Bases V.IOS Press, Amsterdam, 1994, pp. 400-420.

[6] Fagin R. *Multivalued dependencies and a new normal form for relational databases.* ACM TODS 2, 3, Sept. 1977, pp. 262-278.

[7] Fagin R., Beeri C., Howard J. H. *A complete axiomatization for functional and multivalued dependencies in database relations.* Jr. Proc. 1977 ACM SIGMOD Symposium, ed. D. C. P. Smith, Toronto, pp. 47-61.

[8] Fagin R., Vardi Y. V. *The theory of data dependencies: a survey.* Mathematics of Information Processing, Proceedings of Symposia in Applied Mathematics, AMS, 1986, vol. 34, pp. 19-72.

[9] Flach P., Savnik I. *Database dependency discovery: a machine learning approach.* AI Communications,volume 12 (3): 139–160, November 1999.

[10] Flach P., Savnik I. *Discovery of multivalued dependencies from relations.* Intelligent Data Analysis,volume 4 (3,4): 195–211, November 2000.

[11] Ganter, B., Wille R. *Formal Concept Analysis. Mathematical Foundations.* Springer, 1999.

[12] Huhtala Y., Karkkainen J., Porkka P., Toivonen H. *TANE: An Efficient Algorithm for Discovering Functional and Approximate Dependencies.* The Computer Journal 42(2): 100 - 111, 1999.

[13] Khardon R., Roth D. *Reasoning with Models* Artificial Intelligence 87, November 1996, pages 187-213.

[14] Kivinen J., Mannila H. *Approximate inference of functional dependencies from relations.* Theoretical Computer Science 149(1) (1995), 129-149.

[15] Pfaltz, J.L., Taylor, C.M. *Scientific Discovery through Iterative Transformations of Concept Lattices.* Workshop on Discrete Mathematics and Data Mining at 2nd SIAM Conference on Data Mining, Arlington. Pages 65-74. April 2002.

[16] Sagiv Y., Delobel D., Scott Parker D., Fagin R. *An equivalence between relational database dependencies and a fragment of propositional logic.* Jr. J. ACM 28, 3, July 1981, pp. 435-453. Corrigendum: J. ACM 34, 4, Oct. 1987, pp. 1016-1018.

[17] Savnik I., Flach P. *Bottom-up Induction of Functional Dependencies from Relations.* Proc. of AAAI-93 Workshop: Knowledge Discovery in Databases. 1993.

[18] Selman B, Kautz H. *Knowledge compilation and theory approximation* Journal of the ACM Volume 43, Issue 2 (March 1996) Pages: 193 - 224, 1996

[19] Ullman J.D. *Principles of Database and Knowledge-Base Systems.* Computer Science Press, Inc. 1988.

[20] Zaniolo C., Melkanoff M. A. *On the Design of Relational Database Schemata.* ACM TODS 6(1): 1-47 (1981).