

New Closure Operators and Lattice Representations for Multivalued Dependencies and Related Expressions

Jaume Baixeries¹ and José Luis Balcázar¹

¹Dept. Llenguatges i Sistemes Informàtics
Universitat Politècnica de Catalunya
c/ Jordi Girona, 1-3
08034 Barcelona
{jbaixer, balqui}@lsi.upc.edu

Abstract. In Database Theory, Multivalued Dependencies are the main tool to define the Fourth Normal Form and, as such, their inference problem has been deeply studied; two related notions appearing in that study are a syntactical analog in propositional logic and a restriction that maintains to this logic the same relationship as Functional Dependencies do to Horn logic. We present semantic, lattice-theoretic characterizations of such multivalued dependencies that hold in a given relation, as well as similar results for the related notions just mentioned. Our characterizations explain better some previously known facts by providing a unifying framework that is also consistent with the studies of Functional Dependencies.

1 Introduction

Functional Dependencies (FDs) and Multivalued Dependencies (MVDs) are important notions in Database Theory. They help to formalize the necessary conditions on a given relation to ensure, up to various degrees, a reduced amount of redundancy; and they indicate decomposition (more properly, normalization) operations to be performed to obtain such irredundant (normal form) relations.

The most widely studied specific problems about these notions concern how to axiomatize, with a limited number of dependencies, a large set of them, or how to test as efficiently as possible whether a given dependency is a logical consequence of others ([7], [8], [14], [18], [19]).

The less famous notion of Degenerate Multivalued Dependencies (DMVDs) does not play such an important role in database normalization, and has been usually neglected by database design practitioners, but has been studied by theorists. The interest of DMVDs is that they share some features with FDs (they are equality-generating instead of tuple-generating) and some features with MVDs (they have a similar form and analogous syntactical inference rules); then, their study may clarify relationships between FDs and MVDs. In particular, the highly more complex combinatorics of MVDs compared to FDs makes it

nontrivial to extend results for FDs into the MVD realm, and then DMVDs may offer a useful intermediate stepping stone.

In some ways, dependencies (functional or multivalued) behave as certain fragments of propositional logic. The main interest of studying this analogy is double: first, to propose alternative ways to solve inference problems, either for sets of propositional clauses or for sets of dependencies; and, second, to offer alternative semantic proofs of the syntactical correspondences of the (fully analogous) inference calculi available for FDs and MVDs and for their propositional counterparts. This approach thus allows one to simplify, to some extent, the combinatorics contained in the dependencies. More precise explanations of how functional dependencies and multivalued dependencies are related to propositional clauses are given in the next section, and additional details can be found in [18], [19].

Recent progresses in [3] makes it appropriate to revisit the somewhat unsatisfactory results in [6], trying to obtain a better understanding of the comparison-based binarization. We considered there MVD clauses and stated a restricted form of closure under intersection that characterizes the theories axiomatized by such clauses in a form similar to Horn clauses (a full proof can be found in [4]); whereas that result gave a clear characterization for these propositional theories, it did not provide a lattice form due to the fact that, sometimes, the intersection operation may not be of the restricted form that ensures remaining within the theory. Likewise, the lattice form for MVDs did not consist only of propositional models, but they were labeled with the specific pairs of original tuples that gave rise to them, and the intersection had to be extended to the label through a sort of join.

Thanks to the additional knowledge obtained in [1], [2] and [3], we are now able to refine these facts into somewhat more satisfactory lattice-theoretic characterizations. Indeed, the purpose of this paper is twofold: on one hand, to provide a lattice-theoretic characterization of a set of MVD clauses that hold in a propositional theory; it generalizes that of [6] and parallels that of [5] for Horn clauses. On the other hand, to clarify the relationship between MVD lattices and MVD clauses for a larger-than-two-tuples world, by exactly identifying the nodes that must be removed from the binarization in order to get the lattice corresponding to MVDs. This result complements the discussion started in [18] about the fragment of propositional logic induced by MVDs; provides a new method to characterize the set of MVDs that hold in a relation; and contributes a new view of how close the MVD lattice is to the DMVD lattice, with the precise understanding of where the difference lies. We expect this progress to allow for further advances in the study of binary representations for more sophisticated dependencies that fully lack, so far, lattice-theoretic characterizations. A global longer-term aim guiding this research is to elucidate whether there are deep implications between two properties of dependencies, seen in a general view: some of them have Armstrong relations, others do not; and some of them admit lattice-theoretic characterizations, whereas it appears like others do not. We

believe that these two properties have a deep interconnection that is worth to investigate and bring into light.

2 Basic Definitions

Since we will be dealing with sets of data that have different domains (multi-valued and binary) for a set of attributes R , we will denote a set of tuples or a **relation** $r = \{t_1, t_2 \dots t_n\}$ when the attributes are defined over a multi-valued domain, following the usual notation in database research. Likewise we denote a **theory**, that is, a set of propositional models, each a binary tuple, as $T = \{m_1, m_2 \dots m_n\}$ when the domain is binary.

Definition 1. *The **binarization** of r , $BIN(r)$, is the set of binary tuples (theory) that result from comparing all the pairs of tuples attributewise.*

For instance, given a set of tuples $r = \{\{a, b, c, d\}, \{a, b, d, c\}, \{a, c, c, d\}, \{a, c, d, c\}\}$, it would yield $BIN(r) = \{\{1, 0, 0, 0\}, \{1, 0, 1, 1\}, \{1, 1, 0, 0\}\}$. Observe that, being a set, $BIN(r)$ has no duplicates.

2.1 Partitions of attributes

A main combinatorial ingredient of our contributions is given by partitioning the set of attributes into equivalence classes.

Definition 2. *The **partition of attributes** $PART(m_i)$ that results from a model m_i is the partition of R such that each attribute that has a true value in m_i , forms a singleton class, and all the attributes with false value are together in the same class.*

For instance, if $m_i = \{11001\}$, then $PART(m_i) = \{A|B|CD|E\}$. We freely say that $PART(T)$ is the set of partitions for each model of T , with no duplications, of course, even if the same partition arises from several models. For instance, if $T = \{1101, 1011\}$, then $PART(T)$ would be $\{A|B|C|D\}$. The set of all the partitions is denoted \wp .

The key operation on partitions will be here a standard meet operation corresponding to the lattice of partitions; the corresponding join operation is also a usual operation on partitions. Namely, given two partitions, their join is formed by all the classes that can be obtained by pairwise intersecting one class from each, that is, the coarsest partition that is finer than both; whereas the meet is the finest partition that is coarser than both, so that the partial ordering associated to the lattice is the natural ordering where $P \leq P'$ whenever P' is finer than P . More details on alternative ways to define the meet operation can found in [3].

Moreover, through the operation $PART$, we obtain a way of computing a meet operation on models that is different from the standard bitwise intersection. For instance, $1110 \wedge 0011 = 0011$, because $PART(1110) = \{a|b|c|d\}$ and

$PART(0011) = \{ab|c|d\}$, and $\{a|b|c|d\} \wedge \{ab|c|d\} = \{ab|c|d\}$ corresponding to the model 0011. Note that the operator \wedge may not be applicable to some models, in the sense that the meet partition might have more than one nonsingleton class and then there would be no model corresponding to the meet partition (e.g., for 11100 and 00111). Either among partitions or among models, we employ the following notation:

Definition 3. For any given set S , if the operator \wedge is defined on that set, the **closure under meet** of a subset $Q \subseteq S$, denoted $[Q]^\wedge$, is the smallest subset of S that contains Q and is closed under \wedge .

2.2 Clauses and Dependencies

All our expressions have two parts, an antecedent and a consequent, each being a set of attributes or of propositional variables, denoted X , Y , or the like; juxtaposition as in XY denotes union. It is customary to separate them by some sort of double arrow. For now on, we use only the word ‘‘attributes’’ to mean ‘‘attributes or propositional variables’’ whenever this is the necessary interpretation. The various expressions, and their semantics, are as follows.

Definition 4. A **multivalued dependency clause (MVDcl)** $X \rightrightarrows Y$ holds in a theory T if for each $m_i \in T$ if $m_i \models X$ then $m_i \models Y$ or $m_i \models R \setminus XY$. It is, if the X attributes are true, then, the Y attributes are true or the $R \setminus XY$ attributes are true.

Definition 5. A **degenerate multivalued dependency (DMVD)** $X \mapsto Y$ holds in r if whenever two tuples $t_i, t_j \in r$, $t_i[X] = t_j[X]$ implies that $t_i[Y] = t_j[Y]$ or $t_i[R \setminus XY] = t_j[R \setminus XY]$.

Definition 6. A **multivalued dependency (MVD)** $X \twoheadrightarrow Y$ holds in r if whenever two tuples $t_i, t_j \in r$ coincide in X , $t_i[X] = t_j[X]$, it implies that the tuples t'_i, t'_j exist in r , where $t'_i[X] = t'_j[X] = t_i[X]$, $t'_i[Y] = t_i[Y]$, $t'_j[Y] = t_j[Y]$, and conversely $t'_i[R \setminus XY] = t_j[R \setminus XY]$ and $t'_j[R \setminus XY] = t_i[R \setminus XY]$.

Definition 7. The **syntactically equivalent MVD (DMVD)** of a MVDcl $X \rightrightarrows Y$ is $X \twoheadrightarrow Y$ ($X \mapsto Y$); and similarly between MVDs and DMVDs.

It is immediate to check that, if a degenerate multivalued dependency holds in a relation r , so does the syntactically equivalent multivalued dependency. Some common properties for DMVDs, MVDs and MVD clauses are the following:

1. Complementation: If $X \rightarrow Y$ holds, then $X \rightarrow R \setminus Y$ holds.
2. Reflexivity: If $Y \subset X$, then $X \rightarrow Y$ holds.
3. Union of right-hand side: If $X \rightarrow Y$ and $X \rightarrow Y'$ hold, then $X \rightarrow YY'$ holds.

where $X \rightarrow Y$ can be a DMVD, a MVD or a MVD clause. For a proof of these properties the reader can refer to [21] or [2]. For each left-hand side of a MVDcl, a DMVD or a MVD, we can define its **dependency basis** as follows:

Definition 8. *Given a theory T (a relation r), the **MVDcl dependency basis** (**DMVD dependency basis**, **MVD dependency basis**) for a set of attributes X , $DEP(X)$, is a partition of the set of attributes such that for all MVDcl (DMVD, MVD) that have X in their left hand side, they hold in T (in r) if and only if their right hand side may be formed by the union of classes of $DEP(X)$.*

We note that, for a given set of attributes X , all these attributes will appear as singletons in $DEP(X)$ regardless of the kind of dependency basis. This is so because of the reflexivity property that holds in all these dependencies; they are often omitted in this generalized dependency. Additionally, the unions Y mentioned at the end include, as particular cases, the individual classes of the partition $DEP(X)$. The definition of a dependency basis allows us to group all the MVDcl, DMVDs or MVDs that have the same left-hand side in one generalized MVDcl, DMVD or MVD:

Definition 9. *All the MVDcl (DMVD, MVD) that have X in their lefthand side may be represented by a **generalized MVDcl** (DMVD, MVD) $X \Rightarrow DEP(X)$ ($X \mapsto DEP(X)$, $X \rightarrow DEP(X)$).*

In [2] and [3], we defined closure operators on partitions, characterizing all the DMVDs (MVDs respectively) that hold in a relation; the characterizations have the same overall form as Theorem 1 below. They also were proved to derive Armstrong relations for that set of DMVDs (MVDs). An Armstrong relation for a set of dependencies is a relation such that all those dependencies hold in this relation, as well as all the dependencies that are a logical consequence of this set, but the rest of dependencies do not hold. It can be used as an alternative way to characterize a set of dependencies. In this present paper, those closure operators will be called Γ_{DMVD} and Γ_{MVD} respectively.

Another relevant notion about partitions of attributes is that of **matching**, which varies depending whether the partition is considered to be modelling a set of DMVDs or a set of MVDs. We provide here a brief description of this property (more details can be found in [2] and [3]): we say that a partition of attributes $P = \{P_1|P_2|\dots|P_n\}$ matches a set of tuples C (when modelling a set of DMVDs) if all the tuples in C differ only in one class of attributes. If this same partition is considered to be modelling a set of MVDs, we say that P matches a set of tuples C if $C = Pr(P_1, C) \times Pr(P_2, C) \times \dots \times Pr(P_n, C)$, where $Pr(X, C)$ is the projection of C on the attributes X , that is, the set of all tuples obtained from C by removing the values corresponding to attributes not in X .

3 Lattice Characterization of Multivalued Dependency Clauses

It is easy to prove that, given a relation r , a DMVD $X \mapsto Y$ holds in r if and only if the syntactically equivalent MVDcl $X \Rightarrow Y$ holds in $BIN(r)$, as the next proposition states:

Proposition 1. *Let be r a relation, then $X \leftrightarrow Y$ holds in r if and only if $X \Rightarrow Y$ holds in $BIN(r)$.*

According to this result, and due to the fact that the operator Γ_{DMVD} as defined in [2] provides a complete characterization of a set of DMVDs, the lattice that characterizes the MVD clauses that hold in $BIN(r)$ must be the same. However, given a theory T whose lattice we wish to construct, we would need to start by going back into a relation r such that $BIN(r) = T$ and operate on the DMVDs of r ; and care must be exercised to make sure that $BIN(r)$ does not produce additional models. We deem useful to provide a construction of the lattice that goes on directly from T , by means of a new closure operator appropriate to work on models. We indeed provide now an operator that, given a theory, calculates a lattice that characterizes the set of all MVD clauses that hold in that theory, which is the following:

Definition 10. *Fix a theory T . Let $\Gamma_{MVDcl} : \wp \mapsto [PART(T)]^\wedge$ a map such that given a partition of attributes P , returns a partition of attributes P' such that $P \leq P'$ and for all $P'' \in [PART(T)]^\wedge$, if $P \leq P'' \leq P'$ implies that $P'' = P'$.*

Note that it is well-defined since $P \leq P'$ and $P \leq P''$ implies that $P \leq P' \wedge P''$, and the codomain is closed under meet.

Proposition 2. *Γ_{MVDcl} is a closure operator.*

Proof. We prove the three axioms for closure operators.

1. $P \leq \Gamma_{MVDcl}(P)$. Obvious since Γ_{MVDcl} always returns a $P' \in [PART(T)]^\wedge$ such that $P \leq P'$.
2. $\Gamma_{MVDcl}(P) = \Gamma_{MVDcl}(\Gamma_{MVDcl}(P))$. Let $P' = \Gamma_{MVDcl}(P)$ and, besides, $Q = \Gamma_{MVDcl}(P')$ so that $P' \leq Q$. For all $P'' \in [PART(T)]^\wedge$, $P' \leq P'' \leq Q$ implies that $P'' = Q$, and since $P' \in [PART(T)]^\wedge$ and $P' \leq P' \leq Q$, we have that $Q = P'$.
3. $P \leq P' \Rightarrow \Gamma_{MVDcl}(P) \leq \Gamma_{MVDcl}(P')$. Since $P \leq P'$, we have that $P \leq \Gamma_{MVDcl}(P')$, and also $P \leq \Gamma_{MVDcl}(P)$. We then have that

$$P \leq \Gamma_{MVDcl}(P) \wedge \Gamma_{MVDcl}(P') \leq \Gamma_{MVDcl}(P)$$

By the closure under meet, $\Gamma_{MVDcl}(P) \wedge \Gamma_{MVDcl}(P') \in [PART(T)]^\wedge$; by the last condition of the definition, we get $\Gamma_{MVDcl}(P) \wedge \Gamma_{MVDcl}(P') = \Gamma_{MVDcl}(P)$ or, equivalently, $\Gamma_{MVDcl}(P) \leq \Gamma_{MVDcl}(P')$ as desired. ■

Note that $PART(m_i)$ can only express a partition that has only one class that is not a singleton. Then, all the partitions that belong to $PART(T)$ will have only one class that is not a singleton. Partitions with more than one nonsingleton can be obtained through the meet operation. The next theorem proves that Γ_{MVDcl} characterizes all the MVD clauses that hold in a theory T :

Theorem 1. *A generalized MVD clause $X \Rightarrow Y_1 | \dots | Y_m$ holds in T if and only if for $P = \{X_1 | \dots | X_n | Y_1 | \dots | Y_m\}$ and $P' = \{X_1 | \dots | X_n | Y_1 \dots Y_m\}$ we have that $\Gamma_{MVDcl}(P') = P$, where the X_j are the attributes in X .*

Proof. (\Rightarrow) We first note that the fact that a generalized MVD clause $X \Rightarrow Y_1 | \dots | Y_m$ holds in T means that these Y_i constitute the finest possible right hand side for a generalized MVD clause that has X in its left hand side. For each i , each model in T must satisfy $X \Rightarrow Y_i \vee Y'$, where Y' stands for the union of the remaining classes of attributes. This implies that all the zeros of the model are in the same class Y_i , since otherwise it is easy to check that this last condition becomes violated. Conversely, it is immediate to see that, if all the models satisfying X have all their zeros together in one single Y_i , then T satisfies the dependency clause.

We must prove that $P' = \{X_1 | \dots | X_n | Y_1 | \dots | Y_m\}$ belongs to $[PART(T)]^\wedge$, and that no other partition in $[PART(T)]^\wedge$ exists in between P and P' . Consider the theory $T' = \{m \in T \mid m \models X\}$, and let $Q = \bigwedge_{m \in T'} PART(m)$. Clearly all X_i are singletons in all $PART(m)$ for $m \in T'$, and, as just described, all of them have their zeros (the only nonsingleton in $PART(m)$) included in one of the Y_i 's, so that for all such m we have $P' \leq PART(m)$, and the properties of the meet ensure $P' \leq \bigwedge_{m \in T'} PART(m) = Q$. We argue that actually $P' = Q$ so that indeed $P' \in [PART(T)]^\wedge$: assume that $P' < Q$, that is, some class Y_i of P' is split in Q into more than one class. Fix two attributes A and B that belong to Y_i that are in different classes of Q , say Z_A and Z_B respectively; from $P' \leq Q$ as just proved, $Z_A \cup Z_B \subseteq Y_i$. Noting the definition of Q , we see that no model in T' has zeros both in Z_A and in Z_B , since otherwise the meet operation would have joined them into a single class. Then, the observation we made in the previous paragraph tells us that the multivalued dependency clause $X \Rightarrow Z_A$ is true in T , and Z_A should be obtained as a union of Y_i 's, which is not the case. Therefore $P' = Q$, so that $P' \in [PART(T)]^\wedge$.

It remains to prove that there is no other partition $P'' \in [PART(T)]^\wedge$ with $P \leq P'' \leq P'$, except, of course, for P' itself, and this will imply that $\Gamma_{MVDcl}(P) = P'$, as we wish. Assume $P'' \in [PART(T)]^\wedge$, and let $T'' \subseteq T$ such that $P'' = \bigwedge_{m \in T''} PART(m)$. Again all the X_i must be singletons in P'' due to $P \leq P''$, which means that the models in T'' satisfy X , or, equivalently, $T'' \subseteq T'$. It is immediate to see, then, that $\bigwedge_{m \in T'} PART(m) \leq \bigwedge_{m \in T''} PART(m)$ because, due to the way the meet operation works, having at least as many tuples in T' makes the partition at least as coarse as the one corresponding to T'' . That is, we see that $P' \leq P''$; and from the assumption $P'' \leq P'$ we get $P'' = P'$. Taken together, all our consequents prove that $\Gamma_{MVDcl}(P) = P'$.

(\Leftarrow) We prove that if $\Gamma_{MVDcl}(P') = P$, then $X \Rightarrow Y_1 | \dots | Y_n$ holds in T . We use the same simple observation as in the first paragraph of the converse direction. Pick any model $m \in T$ that satisfies X : we will prove that all the zeros of m are in the same Y_i , and this implies that the MVD clause is true in T . This is not difficult now, since whenever m satisfies X all the singletons of P are also singletons of m , that is, $P \leq PART(m)$, and because $PART(m)$ belongs obviously to the lattice, the closure of P , namely P' under our assumption,

must also obey $P' \leq PART(m)$. The only way for this to hold is that the only nonsingleton class of $PART(m)$ is fully included in one of the classes of P' , that is, all the zeros of m are in one single Y_i , and this implies that the MVD clause is true. ■

In fact, completely analogous theorems for DMVDs and MVDs, using the corresponding closure operators instead, were proved in [2] and [3] respectively.

We define $L_{MVDcl}(T)$ as the set of closed partitions of attributes that are closed under Γ_{MVDcl} for T . Since Γ_{MVDcl} is a closure operator, the set $L_{MVDcl}(T)$ is closed under the operator meet. We have seen how to create a lattice to represent the MVD clauses that hold in a theory T and, as we previously mentioned, since the equivalence of a DMVD that holds in a relation and a MVDcl that holds in the binarization of that relation is direct, then we can conclude that the lattices generated by Γ_{MVDcl} and Γ_{DMVD} must be the same.

Theorem 1 could have been proved instead by the following argumentation: By Proposition 1, we have that Γ_{MVDcl} generates the same set of closed partitions in $BIN(r)$ as Γ_{DMVD} in r . Then, to prove Theorem 1 we may also derive it from the following theorem, whose proof resorts heavily to our previous work:

Theorem 2. *Let $P = \{P_1 | \dots | P_n\}$ be a partition of attributes. $P = \Gamma_{MVDcl}(P) \Leftrightarrow P = \Gamma_{DMVD}(P)$.*

Proof. For this proof, we will use the Armstrong relation that can be derived from a set of closed partitions as described in [2].

\Rightarrow $P = \Gamma_{MVDcl}(P)$ implies that in the Armstrong relation for each $P_i \in P$ there will be a pair of tuples that will only differ in P_i , and each pair will be different one from each other. It implies that if BIN is performed between the two tuples of each pair (otherwise it will result in a all zeros model), it will result in a binary model such that all the attributes in P_i will be set to zero, and the rest will be set to one. It induces a set of partitions such that all the attributes not in P_i will be singleton, and those in P_i will be in the same partition. Since it happens for all $P_i \in P$, the meet for all those partitions will be P , which proves that it will be closed under Γ_{DMVD} .

\Leftarrow $P = \Gamma_{DMVD}(P)$ implies that for each $P_i \in P$ that is not a singleton, there is a P' such that all the attributes are singleton except those that are in P_i . For all those models, it would imply that there is a pair of tuples that disagree only in P_i , which constitutes a set of tuples that imply that $P = \Gamma_{MVDcl}(P)$. ■

Theorem 1 and the correspondence between MVD clauses and DMVD's gives us an alternative way to construct a lattice and derive the DMVDs of a relation r : binarizing the tuples in r , and closing under meet the resulting partitions of attributes. This process parallels that of the construction of Armstrong relations for a set of FD's [15] and that of constructing an Armstrong relation for a set of DMVDs in [2].

Corollary 1. $[PART(BIN(r))]^\wedge = L_{MVDcl}(BIN(r)) = L_{DMVD}(r)$.

4 Multivalued Dependency Clauses and Multivalued Dependencies

We just proved that $[PART(BIN(r))]^\wedge = L_{DMVD}(r)$, it is, that binarizing a relation, then transforming these models into partitions of attributes and then closing the resulting models under meet, the set of partitions of attributes so constructed characterizes the DMVDs that hold in r . Our previous paper [2] provided such a construction by a Galois connection operating on partitions and relying strongly on the “matching” condition defined before: now we see that binarization offers a different, more natural way for both obtaining the set of DMVDs that hold in a relation and performing the implication test for DMVDs. Equipped with these facts, we move now to the most relevant notion, MVDs proper: in this section we compare $L_{MVD}(r)$ and $[PART(BIN(r))]^\wedge$. First consider the possibility of having an equivalence of the form $L_{MVD}(r) = [PART(BIN(r))]^\wedge$. This correspondence holds in relations where the sets of attributes that have some values in common are reduced to sets of two tuples (a two-tuples world) as it is proved in [19]. Proposition 3 next states that the MVD lattice is actually included in the DMVD lattice; therefore, if the inclusion is an equality, then the equality $[PART(BIN(r))]^\wedge = L_{MVD}$ holds, otherwise, it does not.

Proposition 3. *Let r be a relation, and let $L_{DMVD}(r)$ and $L_{MVD}(r)$ the lattices that characterize the DMVDs and MVDs respectively that hold in r . Then, $L_{MVD}(r) \subseteq L_{DMVD}(r)$.*

Proof. Fixed r , we prove that if partition P is not in $L_{DMVD}(r)$ then P is not in $L_{MVD}(r)$ either. We use the facts, proved in [2] and [3] respectively, that parallel theorem 1 for DMVDs and MVDs: each application of the corresponding closure operator to a partition P that gives a different partition corresponds to a DMVD, respectively MVD, that holds in the relation given. If $P \notin L_{DMVD}(r)$ then a nontrivial degenerate multivalued dependency with singletons from P as left hand side holds in r ; as indicated just after the definition, if the degenerate multivalued dependency holds then its syntactically equivalent multivalued dependency holds as well; the syntactical equivalence implies that its left hand side is the same and therefore singletons of P , so that the closure operator for MVDs also maps P into a different partition and thus P is not closed in $L_{MVD}(r)$ either. ■

Our next result provides a postprocessing method that eliminates the appropriate models in $BIN(r)$ to obtain the MVD lattice. The method consists of traversing the DMVD lattice top-down, and testing, for each node, the conditions stated in the theorem below; at that point, it is necessary to assume that all the nodes above the current node have been already tested, since we need for the proof the meet of all the nodes in the MVD lattice that sit above the node undergoing the test.

In fact, it is immediate to see that, upon considering a node P , if we take the meet of all the nodes above it in the MVD lattice, we obtain a node $Q \geq P$

which is, in fact, one of them due to the closure under meet; and, of course, to maintain the closure property, in case $Q = P$ then P also belongs to the MVD lattice.

Thus, assume from now on that $P < Q$; since all nodes above P are already correctly placed in the MVD lattice, we know that either P belongs to it, or its closure is exactly Q . Thus, we consider a generalized dependency on the basis of P and Q , and we know from [3] that P is in the lattice if and only if the dependency does not hold. Namely, assuming that $P = \{X_1 | \dots | X_n | Y\}$ with $X = X_1 \dots X_n$ and $Q = \{X_1 | \dots | X_n | Y_1 | \dots | Y_m\}$, P must be removed from the lattice if and only if $r \models X \rightarrow Y_i$ for each i . We prove that considering information local to P and Q suffices.

Theorem 3. *Let $m \in \text{BIN}(r)$ and $P = \text{PART}(m)$, where $P = \{X_1 | \dots | X_n | Y\}$ with $X = X_1 \dots X_n$, and $Q = \{X_1 | \dots | X_n | Y_1 | \dots | Y_m\}$ is the meet of all the nodes above P in $L_{\text{MVD}}(r)$; assume $P < Q$. For each Y_i , consider the related models m_1 and m_2 : m_1 has true the true variables of m and those corresponding to Y_i , and m_2 has all variables true except those in Y_i . It holds that $P \in L_{\text{MVD}}(r)$ if and only if there is a Y_i (with associated m_1 and m_2) and a pair of tuples t_1 and t_2 whose comparison results in m , for which there are no pairs t_3 and t_4 so that the comparisons of t_1 with t_3 and of t_2 with t_4 yield m_1 and the comparisons of t_1 with t_4 and of t_2 with t_3 yield m_2 .*

Proof. One direction is easy. If indeed there is Y_i and the pair of tuples, then it can be readily checked from the properties of m_1 and m_2 that this pair witness that $r \not\models X \rightarrow Y_i$, so the generalized dependency constructed from P and Q does not hold; it follows from the results in [3] that Q cannot be the closure of P . But $\Gamma_{\text{MVD}}(P)$ is in the MVD lattice and, if it was above P , the definition of Q would imply $P < Q < \Gamma_{\text{MVD}}(P)$ which contradicts the properties of the closure operator. Thus the only possibility is $\Gamma_{\text{MVD}}(P) = P$ so that $P \in L_{\text{MVD}}(r)$.

For the converse, in fact all the steps in the previous paragraph are “if and only if”, up to the point that $r \not\models X \rightarrow Y_i$. However, this does not complete the proof since there could be pairs of tuples violating the dependency somewhere else: it remains to argue that t_1 and t_2 violating the implication can be picked so that their comparison is m . In fact we prove more: we prove that all these pairs are there. Fix any pair of tuples whose comparison gives $m' \neq m$: we will prove that they satisfy the conditions for $X \rightarrow Y_i$. Note that what we want to prove, that the $X \rightarrow Y_i$ are satisfied for all Y_i in this case, is the same as proving that the generalized dependency $X \rightarrow Y_1 | \dots | Y_m$ is satisfied: it is the one that would correspond to P in case its closure was Q .

Since $P = \text{PART}(m)$ and X is the union of the singletons of P , that is, the ones of m , the result is immediate whenever m' has a zero among these, because the pair of tuples would not satisfy the antecedent. Thus, we only need to consider $m' > m$. Let $P' = \text{PART}(m')$, which implies that P' has at most one class that is not a singleton and that $P \leq P'$, and suppose first that $Q \leq P'$. Then, Q being coarser, the zeros of m' must be all in the same class in Q and the pair of tuples differ only in one class of Q , which implies that they trivially satisfy the generalized dependency.

Therefore, we suppose now that $Q \not\leq P'$, that is, $P \leq P' \wedge Q < Q$, which implies that $P' \notin L_{MVD}(r)$ since otherwise, due to the closure under meet, Q would not have been the meet of all the part of the MVD lattice above P . This implies that a dependency holds in r associated to P' and $\Gamma_{MVD}(P')$. Its left hand side X' is formed by the singletons of P' , which include those of P , say X_1, \dots, X_k , and some more, say X_{k+1}, \dots, X_n ; call the union of this set $W = W_1W_2$ according to whether they come from Y_i or from its complement.

Regarding its right hand side, made up by the rest of the classes of $\Gamma_{MVD}(P')$, we know that $P \leq P' \leq \Gamma_{MVD}(P')$, which surely is in the MVD lattice, so that $Q \leq \Gamma_{MVD}(P')$ by the definition of Q . Thus, we find that Y_i is W_1 (from the left hand side X') union a union of classes of the right hand side. A pair of tuples corresponding to m' must coincide in $X_1, \dots, X_n = XW_1W_2$, and from the fact that the generalized dependency associated to P' holds we easily obtain the tuples needed to prove that the pair fulfills $X \twoheadrightarrow Y_i$ as well. ■

According to this process we can identify one by one, proceeding from the top down, those nodes in L_{DMVD} that must remain in L_{MVD} and distinguish them from those that must be deleted to construct L_{MVD} from L_{DMVD} .

References

1. Baixeries J. *A Formal Concept Analysis framework to model functional dependencies*. Mathematical Methods for Learning (2004).
2. Baixeries J., Balcázar, J.L. *Characterization and Armstrong relations for Degenerate Multivalued Dependencies using Formal Concept Analysis*. The Third International Conference on Formal Concept Analysis 2005.
3. Baixeries J., Balcázar, J.L. *A Lattice Representation of Relations, Multivalued Dependencies and Armstrong Relations*. 13th International Conference on Conceptual Structures (ICCS'05).
4. Balcázar, J.L. *Query learning of Horn formulas revisited..* To appear in: Computability in Europe 2005..
5. 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).
6. Balcázar, J.L., Baixeries J. *Characterizations of multivalued dependencies and related expressions*. Discovery Science 2004.
7. Beeri C., Fagin R., 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. Beeri C. *On the Membership Problem for Functional and Multivalued Dependencies in Relational Databases*. ACM Transactions on Database Systems, Vol 5, No. 3, September 1980, pages 241-259.
9. Davey B.A., Priestley H.A. *Introduction to Lattices and Order*. Second edition. Cambridge University Press, 1990, 2002.
10. Day A. *A lattice interpretation of database dependencies*. Semantics of Programming Languages and Model Theory (M. Droste and Yu. Gurevich, eds), Gordon and Breach, London.

11. Demetrovics J., Hencsey G., Libkin L., Muchnik I. *Normal Form Relation Schemes: A New Characterization*. Acta Cybernetica. 10(3): 141-164 (1992).
12. Demetrovics J., Libkin L., Muchnik I. *Functional Dependencies in Relational Databases: A Lattice Point of View*. Discrete Applied Mathematics 40(2): 155-185 (1992).
13. Guigues J. L., Duquenne V. *Familles minimales d'implications informatives resultant d'un tableau de données binaires*. Math. Sci. hum., 24(95):5-18, 1986.
14. 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.
15. Fagin R. *Armstrong databases*. Invited paper, Proc. 7th IBM Symposium on Mathematical Foundations of Computer Science, Kanagawa, Japan, May 1982.
16. Ganter, B., Wille R. *Formal Concept Analysis. Mathematical Foundations*. Springer, 1999.
17. Lopes, S., Petit J-M., Lakhil L. *Functional and approximate dependency mining: database and FCA points of view*. Special issue of Journal of Experimental and Theoretical Artificial Intelligence (JETAI) on Concept Lattices for KDD, 14(2-3):93-114, Taylor and Francis, 2002.
18. Sagiv Y. *An algorithm for inferring multivalued dependencies with an application to propositional logic*. Journal of the ACM, 27(2):250-262, April 1980.
19. Sagiv Y., Delobel C., Parker S., Fagin R. *An Equivalence Between Relational Database Dependencies and a Fragment of Propositional Logic*. Journal of the ACM, Vol 28, No 3, July 1981, 435-453.
20. Thalheim B. *Conceptual Treatment of Multivalued Dependencies*. 22nd International Conference on Conceptual Modeling, Chicago, IL, USA, Lecture Notes in Computer Science Springer 2003, pp 363 - 275.
21. Ullman J.D. *Principles of Database and Knowledge-Base Systems*. Computer Science Press, Inc. 1988.