Unified Characterization of Symmetric Dependencies with Lattices

No Author Given

No Institute Given

Abstract. Symmetric dependencies, or MultiValued Dependencies-like, are those dependencies that follow the deduction rules of MultiValued Dependencies (MVD's). These are MVD-clauses, Degenerate MultiValued Dependencies (DMVD's), and MVD's, being the latter ones relevant to the 4th normal form in the relational database model. Previous results have explained how to characterize these dependencies with lattices. However, these characterizations were ad-hoc, and no unified characterization has been provided yet. The purpose of this paper is to present such a common characterization for all these kind of dependencies providing the same framework for all of them, and to extend this generalization to the construction of Armstrong relations.

1 Introduction

Symmetric dependencies (SD's), as stated in [13], or MultiValued Dependencies-like, are those dependencies that follow a certain set of inference rules in order to deduce those dependencies that are logical consequence of a given set of dependencies. These are Multivalued Dependency Clauses (MVD-clauses), Degenerated Multivalued Dependencies (DMVD's) and Multivalued Dependencies (MVD's). Multivalued Dependencies are relevant to the database design process in the relational database systems since they allow a designer to decide whether a database design is in fourth normal form. The remaining dependencies are not that well known among practicioners, but they are of interest for theoretical studies. Although all of them are semantically different, they share the same inference rules and their properties have been studied and compared in [8], [9], [16] and [17].

Demetrovics' work on characterizing a set of functional dependencies with lattices in [14] and [15] was primarily focused on proving that the closed sets of attributes that were obtained from a set of functional dependencies, could be seen as a lattice. That representation in a lattice-like form could show also whether a set of dependencies were in some normal forms. A similar discussion can be found in [12]. Some following results in [20] and in [1] show how that same lattice can be deduced not from a set of functional dependencies, but from a set of tuples. In the first case, transforming the set of tuples into a set of models, and the former, using a Galois connection between the set of the attributes and the set of tuples. These results were based in [23] and [25].

MultiValued dependencies are a more sophisticated type of dependency, and differ from functional dependencies primarily because are tuple-generating, opposed to the equality-generating functional dependencies, and they are of importance in database design, since they show the designer if a relation scheme is in third, Boyce-Codd or fourth normal form (and decompose a relation accordingly if it is not the case). Like functional dependencies, lattice characterization of MVD's have been of interest and studied in different papers. In [13] it has been proved that it is possible to characterize symmetric dependencies with a lattice of the powerset of the attributes, and describes what properties that lattice should have. However, a lattice of the powerset of the attributes is not the only suitable lattice to represent this kind of dependencies. In [5], [4] and [6] it is described how to characterize sets of those dependencies with a lattice of the partition of the attributes.

Apart from characterizing a set of dependencies, lattices have also been able to be the departing point to construct an Armstrong relation for a set of dependencies, when those dependencies were given as a lattice. In [5], [4] there are the results for MVD's and DMVD's.

The purpose of this paper is to present a unified characterization of symmetric dependencies, which consists in:

1. Four axioms for a relation between the set of partition of attributes and the antichains of sets of tuples and two operators between both sets such that their composition is an interior operator.
2. Three such relations, one for each type of dependency, such that the interior operator defined through the axiomatic form over that relation characterizes the set of dependencies (of that type) that hold in the given relation.
3. A general method for obtaining Armstrong relations, based on the same axiomatics, that will generalize particular definitions for DMVD's and MVD's described in [4] and [5].

1.1 Paper Layout

The first section will contain basic definitions used along the paper. The second section will explain the characterization of symmetric dependencies with an interior operator relating sets of sets of tuples and partitions and, hence, a lattice. The following section will contain a generic framework that describes how to construct an Armstrong relation for a lattice characterization as well as the specific case for each dependency on how to apply this framework.

2 Basic Definitions

We will be dealing with a triple $(\mathcal{U}, Dom, \mathcal{R})$, where $\mathcal{U} = \{X_1, \dots, X_n\}$ is a set of attributes, $Dom = \{Dom(X_1), \dots, Dom(X_n)\}$ is the set of domains of values for each attribute X_i, and a **relation** $\mathcal{R}$ is a set of tuples $\mathcal{R} = \{t_1, \dots, t_n\}$, such that $\mathcal{R} \subseteq Dom(X_1) \times \cdots \times Dom(X_n)$, where $r(X_i) \in Dom(X_i)$ for all i.

Definition 1. *Let $X \subseteq \mathcal{U}$ and $R \subseteq \mathcal{R}$. We define the* **projection** *of R onto X as $\pi_X(R) := \{t(X) : t \in R\}$, where $t(X)$ is the usual restriction of the mapping t on X.*

Definition 2. *Let X, Y two disjoint subsets of attributes and $R, R' \subseteq \mathcal{R}$. We define the* **cartesian product** *$\pi_X(R) \times \pi_Y(R') := \{t | t(X) \in R \text{ and } t(Y) \in R'\}$, as a set of tuples on $X \cup Y$.*

When all the domains in Dom are binary, then we will be dealing with a triple $(\mathcal{U}, Dom, \mathcal{T})$, where a **theory** $\mathcal{T}$ is a set of propositional models $\mathcal{T} = \{t_1, \ldots, t_n\}$, such that $\mathcal{T} \subseteq \{0, 1\}^n$.

Given a set $\mathcal{U} = \{X_1, \ldots, X_n\}$, we define $Part(\mathcal{U})$ as the set of all possible partitions of $\mathcal{U}$. The notation of a partition P will be as follows: $P = \{P_1 | \ldots | P_n\}$, where each P_i is a class of attributes.

2.1 Orders

Let $P = \{P_1, \ldots, P_n\}$ and $Q = \{Q_1, \ldots, Q_m\}$ be two finite sets. We define the following relations on $Part(P)$ and $\wp(\wp(Q))$:

1. Let $P, P' \in Part(P)$, we say that $P \preceq P'$ if each class in P is contained in some class of P'.
2. Let $Q, Q' \in \wp(\wp(Q))$, we say that $Q \trianglelefteq Q'$ if each set in Q is contained in some set of Q'.

Of course reversing the partial order relation we still have a partial order that preserves all the combinatorics, except for the orientation (we say polarity) of the properties. We find ourselves in the following quandary: whereas it is standard to use closure operators on lattices to discuss the finding of implications in them, to follow this usage we are forced to use a polarity that diverges from the usual one that can be found in the literature. We have chosen to follow literally the polarity of the set $Part(P)$ as per [22], and that of $\wp(\wp(Q))$ is the one that is consistent with the same source. As a consequence, the composition of new functions in the next sections, from and to those sets will result in a interior operator instead of a closure operator. Needless to say that a change on the polarity of the above mentioned sets would change the interior operator into a closure operator. In any case, we would like to remark that, regardless of the polarity of those sets, the symmetry between closure and interior operators guarantees that reversing the polarity will not change the basic properties of the operators that will be defined.

Proposition 1. *The relation $\preceq$ is an order ([22]).*

The relation $\trianglelefteq$ is a quasi-order because it is not antisymmetric. However, if we restrict the set $\wp(\wp(Q))$ to the inclusion-wise antichains, then, we have an order. This set will be called **simple parts of parts of a set** Q (or, also, the set of antichains), and will be formalized as $\wp_S(\wp(Q))$. In graph theory, this set is known as the set of simple hypergraphs.

Proposition 2. *$\trianglelefteq$ on the set $\wp_S(\wp(Q))$ is an order.*

Proof. In the Appendix A.

Definition 3. *The operations union ($\cup$) and intersection ($\cap$) on the set $\wp(\wp(Q))$ are:*

1. $C \cup C' = \{\mu | \mu \in C$ ***or*** $\mu \in C'\}$
2. $C \cap C' = \{\mu \cap \mu' | \mu \in C$ ***and*** $\mu' \in C'\}$

The set $\wp(\wp(Q))$ is closed under the operators $\cup$ and $\cap$.

Definition 4. *The funcion $simple : \wp(\wp(Q)) \mapsto \wp_S(\wp(Q))$ is defined as $simple(C) = \{\mu | \mu \in C$ ***and*** $\nexists \mu' \in C : \mu \subset \mu'\}$.*

Proposition 3. *$simple(C) \trianglelefteq C \trianglelefteq simple(C)$.*

Definition 5. *The operations meet and join for the set $\wp_S(\wp(Q))$ are:*

1. $C \wedge C' = simple(C \cap C')$
2. $C \vee C' = simple(C \cup C')$

Proposition 4. *Given the set $P = \{P_1, \ldots, P_n\}$, the set $Part(P)$ equipped with the order relation $\preceq$, is a complete lattice. The top element is $\{P\}$ and the bottom element is $\{P_1 | \ldots | P_n\}$. ([22])*

The reader can refer to [22] for more information on the lattice of the partitions of a set, but we would like to remind that $P \wedge P'$ will result in a partition such that two elements will be in the same class in $P \wedge P'$ if and only if they are in the same class in P and in P'. $P \vee P'$ will result in the finest partition such that every class in P and in P' will be contained in some class of $P \vee P'$.

Proposition 5. *Given the set $Q = \{Q_1, \ldots, Q_n\}$, the set $\wp_S(\wp(Q))$, equiped with the order relation $\trianglelefteq$, is a complete lattice. The top element is $\{Q\}$ and the bottom element is $\{\emptyset\}$.*

Proof. In the Appendix A.

2.2 Symmetric Dependencies

Let X, Y, Z be sets of attributes.

Definition 6. *A* **multivalued dependency clause** *$X \twoheadrightarrow Y$ holds in a theory iff whenever X is true, then, Y is true or Z is true, where $Z = \mathcal{U} \setminus (X \cup Y)$.*

Definition 7. *A* **degenerate multivalued dependency** *$X \Rrightarrow Y$ holds in a relation iff whenever $t_1(X) = t_2(X)$ implies $t_1(Y) = t_2(Y)$* ***or*** *$t_1(Z) = t_2(Z)$, where $Z = \mathcal{U} \setminus (X \cup Y)$.*

Definition 8. *A* **multivalued dependency** $X \twoheadrightarrow Y$ *holds in a relation* $\mathcal{R}$ *iff whenever two tuples* t_1, t_2 *such that* $t_1(X) = t_2(X)$ *are in* $\mathcal{R}$*, there is a tuple* t_3 *in* $\mathcal{R}$ *such that* $t_1(XY) = t_3(XY)$ *and* $t_3(\mathcal{U} \setminus (X \cup Y)) = t_2(\mathcal{U} \setminus (X \cup Y))$.

Some common properties for symmetric dependencies are the following:

1. Reflexivity: If $Y \subseteq X$, then $X \rightarrow Y$ holds.
2. Complementation: If $X \rightarrow Y$ holds, then $X \rightarrow \mathcal{U} \setminus Y$ holds.
3. Union of right-hand side: If $X \rightarrow Y$ and $X \rightarrow Y'$ hold, then $X \rightarrow Y \cup Y'$ holds.
4. Intersection of right-hand side: If $X \rightarrow Y$ and $X \rightarrow Y'$ hold, then $X \rightarrow Y \cap Y'$ holds.

Definition 9. *Let* $X \subseteq \mathcal{U}$ *and let* Σ *be a set of symmetric dependencies. The* **dependency basis** *of* X *in* Σ*,* $DB(X)$ *is the coarsest partition of* $\mathcal{U}$ *such that for all non empty* Y*'s for which* $X \twoheadrightarrow Y$ *holds,* Y *is a union of classes of* $DB(X)$*.*

Because of the reflexivity property for SD's, we have that in $DB(X)$ the attributes in X will necessarily be singletons.

3 Interior Operators for Characterizing Symmetric Dependencies

We have two finite complete lattices P, Q, with an order $\succeq, \trianglerighteq$ in each of them as well as a binary relation:

Definition 10. *A binary relation* $*$*:* $P \times Q \mapsto \{0, 1\}$ *is a* ***symmetric dependence matching*** *(briefly a matching, and we just read* $p * q$ *as* p *matches* q*) if the following four axioms hold, for all* $p, p' \in P$ *and all* $q, q' \in Q$*:*

1. $p' \succeq p$ *and* $p * q \longrightarrow p' * q$.
2. $p * q$ *and* $p' * q \longrightarrow (p \wedge p') * q$.
3. $\bigvee\{q' | p' * q'\} \trianglerighteq \bigvee\{q' | p * q'\}$ *and* $p'' * \bigvee\{q' | p' * q'\} \longrightarrow p'' * \bigvee\{q' | p * q'\}$.
4. $p * q$ *and* $p * q' \longrightarrow p * (q \vee q')$.

We see later that such combinatorial objects exist and are interesting. We now define the following functions:

Definition 11. *The functions* $\varphi : P \mapsto Q$ *and* $\psi : Q \mapsto P$ *are:*

$$\varphi(p) = \bigvee\{q' | p * q'\}$$

$$\psi(q) = \bigwedge\{p' | p' * q\}$$

Note that, according to the definitions of φ and ψ, Definition 10.3 can be rephrased as: $\varphi(p') \trianglerighteq \varphi(p)$ and $p'' * \varphi(p')$ implies that $p'' * \varphi(p)$.

Proposition 6. *Definitions in 10 imply:*

1. $p * \varphi(p)$.
2. *If* $p' \succeq p$ *then* $\varphi(p') \trianglerighteq \varphi(p)$.
3. $\psi(q) * q$.
4. *If* $\varphi(p') \trianglerighteq \varphi(p)$ *then* $\psi(\varphi(p')) \succeq \psi(\varphi(p))$.

Proof. 1. Obvious by associativity and finiteness.

2. As just stated, $p * \varphi(p)$ and by Definition 10.1, $p' * \varphi(p)$. Hence, $\varphi(p) \in \{q|p' * q\}$, and then, $\varphi(p) \trianglelefteq \bigvee\{q|p' * q\} = \varphi(p')$.
3. Obvious by associativity and finiteness.
4. Let p'' be $\bigwedge\{p|p * \varphi(p')\} = \psi(\varphi(p'))$. As just stated, we have that $p'' * \varphi(p')$, and, by Definition 10.3, $p'' * \varphi(p)$. Then, $p'' \in \{p|p * \varphi(p)\}$ and $p'' \succeq \bigwedge\{p|p * \varphi(p)\}$, that is, $\varphi(\psi(p')) \succeq \varphi(\psi(p))$. ∎

Proposition 7. *The property* $\varphi(p') \trianglerighteq \varphi(p) \longrightarrow \psi(\varphi(p')) \succeq \psi(\varphi(p))$ *is equivalent to the property in Definition 10.3.*

Proof. Assume $\varphi(p') \trianglerighteq \varphi(p)$ and $p'' * \varphi(p')$, that is, $p'' \in \{p|p * \varphi(p')\}$, from where we know $p'' \succeq \psi(\varphi(p'))$. Since $\psi(\varphi(p')) \succeq \psi(\varphi(p))$, then $p'' \succeq \psi(\varphi(p))$. By Proposition 6.3, we have that $\psi(\varphi(p)) * \varphi(p)$, and by 10.1, we have that $p'' * \varphi(p)$. ∎

Theorem 1. $\Gamma = \psi \cdot \varphi$ *is an interior operator.*

Proof. 1. $p \succeq \Gamma(p)$. By Proposition 6.1 we have that $p * \varphi(p)$, and hence, $p \in \{p'|p' * \varphi(p)\}$ and then, $p \succeq \bigwedge\{p'|p' * \varphi(p)\} = \psi(\varphi(p)) = \Gamma(p)$.

2. $p' \succeq p \longrightarrow \Gamma(p') \succeq \Gamma(p)$. If $p' \succeq p$, by Proposition 6.2 $\varphi(p') \trianglerighteq \varphi(p)$, and by Definition 10.3 we have that $\psi(\varphi(p')) \succeq \psi(\varphi(p))$.
3. $\Gamma(p) = \Gamma(\Gamma(p))$. By Theorem 1 we have that $p \succeq \Gamma(p)$, and then, by Proposition 6.2, $\varphi(p) \trianglerighteq \varphi(\Gamma(p))$. We also have that $\Gamma(p) = \bigwedge\{p'|p' * \varphi(p)\}$. Then, it follows by Proposition 6.4 that $\Gamma(p) * \varphi(p)$. Hence, $\varphi(p) \in \{q|\Gamma(p) * q\}$, and then, $\varphi(p) \trianglelefteq \bigvee\{q|\Gamma(p) * q\}$, which is equivalent to $\varphi(p) \trianglelefteq \varphi(\Gamma(p))$. Then, we have that $\varphi(p) = \varphi(\Gamma(p))$, and $\psi(\varphi(p)) = \psi(\varphi(\Gamma(p)))$, that is: $\Gamma(p) = \Gamma(\Gamma(p))$. ∎

3.1 Lattice Characterization of Symmetric Dependencies

Different characterization for symmetric dependencies have been proved in [6], [4] and [5] with a closure operator that, in the second and third cases (for DMVD's and MVD's) was derived from a Galois connection and the first was an ad-hoc definition. The fact is that the definition of those closure operators relied on a binary relation that was set between partitions of attributes and sets of tuples in the first case, and partitions of attributes and antichains of sets of tuples in the former cases. In the previous section we have defined a pair of functions between two sets based on a binary relation, whose composition gives rise to an interior operator. In this section, we define different kinds of relations between partitions of attributes and set of tuples or antichains of sets of tuples, that follow the properties of Definition 10 in such a way that that the semantics for each kind of dependency is condensed in the definition of the relation, which, in fact, allows to use the same definition of the functions that will compose the interior operator. We prove that the characterization that has been defined in the previous section generalizes the definitions of the closure operators that were defined in previous papers.

Some comments must be considered in order to overcome some slightly conceptual changes that have been made in this paper. First, it is necessary to recall that in those previous papers the polarity in the lattice of $Part(\mathcal{U})$ and $\wp_S(\wp(\mathcal{R}))$ gave rise to a closure operator, whereas in this paper an interior operator has been used thoughout. However, due to the symmetry between both kinds of operators, the characterizations given in this paper, given in terms of an interior operator, and those given in prevous papers, in terms of a closure operator, remain symmetric and are mutually interchangeable.

Secondly, if we change Definition 10.3 by a stronger condition,

Definition 12. *$q' \trianglerighteq q$ and $p * q' \longrightarrow p * q$.*

then, φ and ψ are a Galois connection. The proof is given in the Appendix B. It is also straightforward that Definition 12 implies Definition 10.3, and then, to describe the matching relation for MVD-clauses and DMVD's we will use Definition 12 instead of Definition 10.3, which implies that the pair of functions φ and ψ will be a Galois connection, and, therefore, their composition in either way will be an interior and a closure operator. However, when defining the matching relation for MVD's, Definition 10 will remain intact, and, then, the only relevant property of the composition of both functions in one specific way will be the fact that it forms an interior operator.

We recall that P will stand for partition of attributes, and P_i will be class of attributes. For antichains of tuples we will use R, and R_i will be sets of tuples and T will we a set of models.

Characterization of MVD-clauses The matching relation for characterizing a set of MVD-clauses that hold in a relation is a map between the set of partitions of the attributes and sets of models. The formal definition is as follows:

Definition 13. *The relation **mvdcl-matches** $\underset{mvdcl}{*}$: $\{Part(\mathcal{U}) \times \wp(\mathcal{T})\} \mapsto \{0,1\}$ is defined as follows: a partition of attributes $P = \{P_1|\ldots|P_n\}$ **mvdcl-matches** a set of models T ($P \underset{mvdcl}{*} T$) if and only if different models have false values in only one class of attributes: $P \underset{mvdcl}{*} T \Leftrightarrow \forall m \in T : m(P_i) = false$ **and** $m(P \setminus P_i) = true$.*

Note that it is not necessary that all the tuples in the set have all their false values in the same class.

Proposition 8. *The binary relation* $\underset{mvdcl}{*}$ *enjoys the properties of Definition 10.*

Proof. 1. Let $T \subseteq \mathcal{T}$ be a set of models matched by P, and let $m \in T$. Since this class is in T, it means that it has one class set to false, let it be P_i. Now, since $P' \succeq P$, all the attributes in the class P_i will be in the same class in P', and hence, m will have only one class set to false according to the partition P'.

2. Let $T \subseteq \mathcal{T}$ be a set of models matched by P and P'. Let X be the attributes that are false in m. For each model $m \in T$ we have that there is one class set to false according to P and one class set to false according to P'. Then, it implies that $X \subseteq P_i \cap P'_j$, where $P_i \in P$ and $P'_j \in P'$. Precisely, according to the definition of $\wedge$ in the set $Part(\mathcal{U})$ we have that the class $P_i \cap P'_j$ will be present in $P \wedge P'$, which means that the attributes in X will remain in the same class, and thus, guaranteeing that the model m will have only one class set to false according to $P \wedge P'$.
3. It is obvious that if we extract a subset of models from a set of models T, the remaining models will have only one class set to false according to P and hence, the remaining set will be also be matched by P.
4. $T \vee T'$ is the union of both sets. For each model in any of both sets, it will have one class set to false according to P. Hence, their union will be matched by P. ∎

We now define $\Gamma_{mvdcl}(P) = \psi_{dmvd}(\phi_{mvdcl}(P))$ for the functions:

$$\phi_{mvdcl}(P) = \bigvee\{T' | P \underset{mvdcl}{*} T'\}$$

$$\psi_{mvdcl}(T) = \bigwedge\{P' | P' \underset{mvdcl}{*} T\}$$

Although Γ_{mvdcl} is slightly different from that defined in [6], it is easy to see that they are completly equivalent. Then, we have that the following characterization for a set of MVD-clauses:

Theorem 2. *A MVD-clause* $X \twoheadrightarrow Y$ *holds in a theory* $\mathcal{T}$ *if and only if* $\Gamma_{mvdcl}(P) = \Gamma_{mvdcl}(P')$*, where* $P = \{X_1|\ldots|X_n|Y \cup Z\}$ *and* $P' = \{X_1|\ldots|X_n|Y|Z\}$ *and* $Z = \mathcal{U} \setminus (X \cup Y)$*.*

Proof. In [6].

Characterization of DMVD's

Definition 14. *The relation* ***dmvd-matches*** $\underset{dmvd}{*}$: $Part(\mathcal{U}) \times \wp_S(\wp(\mathcal{R})) \mapsto \{0,1\}$ *is defined as follows:*

$$P \underset{dmvd}{*} R \Leftrightarrow \forall R_i \in R : \exists! P_i \in P : \forall t_i, t_j \in R_i : t_i(P_i) \neq t_j(P_i)$$

It is: for each set R_i *in* R*, the tuples in* R_i *disagree in their values in exactly one class of* P*.*

Proposition 9. *The relation **dmvd-matches** enjoys the properties of Definition 10.*

Proof. 1. $P \underset{dmvd}{*} R$ and $P \underset{dmvd}{*} R' \longrightarrow P \underset{dmvd}{*} (R \vee R')$.
We take a set $R_i \in R \vee R'$, which according to the definition of the operation join in $\wp_S(\wp(\mathcal{R}))$ will be in R or in R'. Since $P \underset{dmvd}{*} R$ and $P \underset{dmvd}{*} R'$, then, R_i will differ in only one class of P. This will happen to all the sets in R, and then, we have that $P \underset{dmvd}{*} (R \vee R')$.
2. $P' \succeq P$ and $P \underset{dmvd}{*} R \longrightarrow P' \underset{dmvd}{*} R$. By definition of the order in $Part(\mathcal{U})$ we have that $\forall P_i \in P : \exists P_j \in P' : P_i \subseteq P_j$. We take a $R_i \in R$. Since $P \underset{dmvd}{*} R$, all the tuples in R_i disagree in only one class of attributes, let it be $P_i \in P$. Since $P' \succeq P$, the class P_i will be contained in a class $P_j \in P'$ such that $P_i \subseteq P_j$, and then, all the attributes in which all the tuples in R_i disagree will be contained in one single class in P'. It will happen for all $R_i \in R$, which yields that $P' \underset{dmvd}{*} R$.
3. $P \underset{dmvd}{*} R$ and $P' \underset{dmvd}{*} R' \longrightarrow (P \wedge P') \underset{dmvd}{*} (R \wedge R')$. We take a $R_i \in R$. By definition of the operation meet for $\wp_S(\wp(\mathcal{R}))$ we have that $R_i = R_1 \cap R_2$ such that $R_1 \in R$ and $R_2 \in R'$. On the other hand, all the attributes X in which all the tuples in R_1 disagree will be contained in one single class $P_i \in P$ and for all the tuples in R_2, the attributes X' in which they disagree will be contained in $P_j \in P'$. Then, the tuples in R_i will disagree in $X \cup X'$, which will have to be contained in both P_i and P_j, and, then, in their intersection. By definition of the operation meet on the set $Part(\mathcal{U})$, the set of attributes $P_i \cap P_j \in P \wedge P'$, and will contain all the attributes in which all the tuples in R_i disagree. This will happen to all $R_i \in R$, and, then, we have that $(P \wedge P') \underset{dmvd}{*} (R \wedge R')$.

∎

We now define $\Gamma_{mvdcl}(P) = \psi_{dmvd}(\phi_{dmvd}(P))$ for the functions:

$$\phi_{dmvd}(P) = \bigvee \{R' | P \underset{dmvd}{*} R'\}$$

$$\psi_{dmvd}(R) = \bigwedge \{P' | P' \underset{dmvd}{*} R\}$$

Again, both the operator itself and the following characterization of the DMVD's are slightly different from in [4], but it is easy to see they are equivalent:

Theorem 3. *A degenerated multivalued dependency $X \twoheadrightarrow Y$ holds in a relation $\mathcal{R}$ if and only if $\Gamma_{dmvd}(P) = \Gamma_{dmvd}(P')$, where $P = \{X_1 | \dots | X_m | Y \cup Z\}$ and $P' = \{X_1 | \dots | X_m | Y | Z\}$, where all X_i are singletons and $Z = \mathcal{U} \setminus (X \cup Y)$.*

Proof. In [4].

Characterization of MVD's

Definition 15. *Let $P = \{P_1|P_2|\dots|P_n\} \in \mathcal{P}$. The relation **mvd-matches** : $\underset{mvd}{*} Part(\mathcal{U}) \times \wp_S(\wp(\mathcal{R})) \mapsto \{0,1\}$ is defined as follows:*

$$\forall R_i \in R : R_i = \pi_{P_1}(R_i) \times \cdots \times \pi_{P_n}(R_i)$$

This time it will be handy, for the proof that the axioms are satisfied, to have already in place the functions:

$$\phi_{mvd}(P) = \bigvee\{R'|P \underset{mvd}{*} R'\}$$

$$\psi_{mvd}(R) = \bigwedge\{P'|P' \underset{mvd}{*} R\}$$

Proposition 10. *The relation **mvd-matches** enjoys the properties of Definition 10.*

Proof. 1. $P' \succeq P$ and $P \underset{mvd}{*} R \longrightarrow P' \underset{mvd}{*} R$. A reordering of the universe of attributes suffices to see this, as a form of associativity of the cartesian product.
2. $P \underset{mvd}{*} R$ and $P' \underset{mvd}{*} R \longrightarrow P \wedge P' \underset{mvd}{*} R$. Let $P'' = P \wedge P'$, and $P'' = \{P''_1|\dots|P''_s\}$. Within each class of P, the classes of $P \wedge P'$ are the traces of the classes of P' and thus there is no difference between $P \wedge P'$ and P' itself. Therefore, considering $\pi_{P_i}(R)$ for $P_i \in P$, since $P' \underset{mvd}{*} R$, $\pi_{P_i}(R)$ coincides indeed with the product of all the projections of R according to P'; and the complete cartesian product of these projections again is R since $P \underset{mvd}{*} R$. Taken together, on the one hand they reconstruct R and, on the other, they form the product of all the projections according to $\{P''_1|\dots|P''_s\}$; that is, $P'' \underset{mvd}{*} R$.
3. Perhaps this is, in fact, possibly the subtlest point and most delicate argument, and one key contribution of this paper. We use here Proposition 7, whereby we prove $\varphi(p') \trianglerighteq \varphi(p) \longrightarrow \psi(\varphi(p')) \succeq \psi(\varphi(p))$ in order to argue this property. To reduce the notation load we use locally here φ and ψ instead of ϕ_{mvd} and ψ_{mvd}. Thus suppose $\varphi(P') \trianglerighteq \varphi(P)$, that is, each set of tuples mvd-matched by P is included in some set mvd-matched by P'. Assume that $\psi(\varphi(P')) \succeq \psi(\varphi(P))$ fails: there must be a class P_i of $\psi(\varphi(P))$ split by the classes of $\psi(\varphi(P'))$, that is, there must be attributes A and B in P_i falling into different classes P'_A and P'_B of $\psi(\varphi(P'))$. Let $Z = \mathcal{U} - P_i$. We will construct a new partition $P'' = \{P''_A, P''_B, Z'', Z\}$ by decomposing P_i into three classes, and ensuring that $A \in P''_A$ and $B \in P''_B$ so that actually $P'' \wedge \psi(\varphi(P))$ is finer than $\psi(\varphi(P))$.
Assume briefly that all q in $\varphi(P)$ are matched by P'': then $\psi(\varphi(P))$ would not be the finest partition matching $\varphi(P)$, which would contradict the very definition of ψ. Therefore, there is some $q \in \varphi(P)$ which is not mvd-matched

by P''. Fix that q for the rest of the proof. From $q \in \varphi(P)$ we know that $\{q\}$ is mvd-matched both by P and $\psi(\varphi(P))$.
Thus the two-class partition $\{P_i|Z\}$, being coarser than $\psi(\varphi(P))$, also does.
We can now reorder $\mathcal{U}$ so that $q = \pi_{P_i}(q) \times \pi_Z(q) = \bigcup_{v \in \pi_Z(q)}(q_v)$ where $q_v = \pi_{P_i}(q) \times \{v\}$. Clearly $q_v \subseteq q$, and from the hypothesis $\varphi(P') \trianglerighteq \varphi(P)$ we obtain some $q' \in \varphi(P')$ such that $q_v \subseteq q \subseteq q' = \prod_{P'_j \in \psi(\varphi(P'))} \pi_{P'_j}(q')$, the last equality coming from the fact that $q' \in \varphi(P')$ and is thus matched also by $\psi(\varphi(P'))$.
Let $Z' = \mathcal{U} - (P'_A \cup P'_B)$, and define our partition P'' as follows: $P''_A = P'_A - Z$, $P''_B = P'_B - Z$, $Z'' = Z' - Z$, and $P'' = \{P''_A|P''_B|Z''|Z\}$. Note that it is independent of the choice of q, and that it indeed refines P_i into three classes as indicated.
Observing that all the tuples of q_v have the same values along Z, namely v, it follows that (again reordering $\mathcal{U}$)

$$\pi_{P''_A}(q_v) \times \pi_{P''_B}(q_v) \times \pi_{Z''}(q_v) \times \pi_Z(q_v) = \pi_{P''_A}(q_v) \times \pi_{P''_B}(q_v) \times \pi_{Z''}(q_v) \times \{v\} \subseteq q' \subseteq \mathcal{R}$$

and we can take the union over $v \in \pi_Z(q)$ and get

$$\pi_{P''_A}(q) \times \pi_{P''_B}(q) \times \pi_{Z''}(q) \times \pi_Z(q) \subseteq \mathcal{R}$$

However, we know that q was chosen so that it was not mvd-matched by P'', and this implies a proper inclusion

$$q \subset \pi_{P''_A}(q) \times \pi_{P''_B}(q) \times \pi_{Z''}(q) \times \pi_Z(q)$$

so that q is not a maximal set of tuples matched by $\psi(\varphi(P))$. But this is a contradiction to the fact that $q \in \varphi(P) = \varphi(\psi(\varphi(P)))$ just as in the proof of Theorem 1.

4. $P \underset{mvd}{*} R$ and $P \underset{mvd}{*} R' \longrightarrow P \underset{mvd}{*} R \vee R'$.
We take a $R_i \in R \vee R'$. By definition of the operation join in the set $\wp_S(\wp(\mathcal{R}))$ we have that $R_i \in R$ **or** $R_i \in R'$, in both cases, it means that $R_i = \pi_{P_1}(R_i) \times \cdots \times \pi_{P_n}(R_i)$. This will be so for all the sets of tuples in $R \vee R'$, which yields $P(\underset{mvd}{*} R \vee R')$.

∎

We now define $\Gamma_{mvd}(P) = \psi_{mvd}(\phi_{mvd}(P))$

As in the previous two cases, the following characterization of the MVD's is slightly different as in [5], but this characterization is equivalent and, in fact, solves some technical problems in the purported Galois connection defined in [5]:

Theorem 4. *A multivalued dependency $X \twoheadrightarrow Y$ holds in a relation r if and only if $\Gamma_{mvd}(P) = \Gamma_{mvd}(P')$, where $P = \{X_1|\ldots|X_m|Y \cup Z\}$ and $P' = \{X_1|\ldots|X_m|Y|Z\}$, where all X_i are singletons and $Z = \mathcal{U} \setminus (X \cup Y)$.*

Proof. In [5].

4 Armstrong Relations for Symmetric Dependencies

It is not implied by the Definition 10 that the partition of attributes $\{\mathcal{U}\}$ should match any special set of tuples or antichain of tuples. However, and according to the definition of the particular matching relations for all three different kinds of symmetric dependencies, it follows that $\{\mathcal{U}\}$ matches any possible set or antichain of tuples, it is: $\{\mathcal{R}\}$ for MVD's and DMVD's, and $\{\mathcal{T}\}$ for MDV-clauses.

In [4] and [5] were presented two different ad hoc methods for constructing Armstrong relations for the set of dependencies represented by a lattice of partitions of attributes. In this section we explain a unified method for symmetric dependencies that are defined over a set of multivalued, it is: DMVD's and MVD's.

First, we need a new condition on the binary relation matches:

Property 1. Let $R_1, R_2 \subseteq \mathcal{R}$ be two non-empty subsets of tuples such that there is no $t \in \mathcal{R}$ such that for $X, X' \in \mathcal{U}$ and $t_1 \in R_1$ and $t_2 \in R_2$ we have that $t(X) = t_1(X)$ and $t(X') = t_2(X')$. Then, there will be no P (except $\{\mathcal{U}\}$) that will match an antichain of tuples that will contain a set of tuples such that there will be at least one tuple from R_1 and one from R_2.

It is easy to prove that the matching relations $\underset{dmvd}{*}$ and $\underset{mvd}{*}$ enjoy the property stated in Definition 1.

Proposition 11. $\underset{dmvd}{*}$ *enjoys property 1.*

Proof. Let us assume that a partition of attributes P with, at least, two classes, matches an antichain of tuples that contains a set of tuples that, at least, has one tuple t_1 from R_1 and one tuple t_1 from R_2. Since there is no tuple in $\mathcal{R}$ that has values common to those in R_1 and R_2, all the values of each attribute will be different. If there are at least two classes in P, then, there will be two classes of attributes that will be different comparing tuples in the same class, which contradicts the definition of $\underset{dmvd}{*}$. ∎

Proposition 12. $\underset{mvd}{*}$ *enjoys property 1.*

Proof. Let us suppose that we have a partition P with, more than one class. Then, for each set of tuples R_i, it must be expressed as: $\Pi_{P_1}(R_i) \times \cdots \times \Pi_{P_n}(R_i)$ for $n \geq 1$. Let us suppose that there is one P_i that contains at least one tuple from R_1 and one from R_2. Then, regardless of whether the values from R_1 and R_2 fall in the same class or not, it is obvious that R_i will contain at least a tuple with values from R_1 and R_2, but this is not the case because of Definition 1. ∎

Definition 16. *Given a lattice $L \in Part(\mathcal{U})$, we define $cover(L)$ as the set of join-irreducible elements of L.*

Definition 17. *Given a lattice $L \in Part(\mathcal{U})$, a binary relation $*$ between $Part(\mathcal{U})$ and $\wp_S(\wp(\mathcal{R}))$, and an interior operator Γ formed by the composition of a Galois connection based on the relation $*$ as defined in 10, we define as $tuples(P)$, where $P \in L$, a set of tuples such that:*

1. *$P * tuples(P)$.*
2. *Given a relation $r = tuples(P)$, we have that $\Gamma(P) = P$ in r.*

In [4] and in [5] it is defined the function *tuples* for DMVD's and MVD's.

Proposition 13. *For all P' such that $P' \not\geq P$, we* ***do not*** *have that $P' * tuples(P)$ when $\mathcal{R} = tuples(P)$.*

Proof. Let us suppose that it is so: then, by Definition 10.2 we have that $(P \wedge P') * tuples(P)$, and, since $\{tuples(P)\} = \{\mathcal{R}\}$ is the largest antichain of tuples that can be matched, then, we would have that P would not be closed in $tuples(P)$, contradicting Definition 17.2. ∎

Proposition 14. *Let $\mathcal{R} = \bigcup_{R_i \in cover(L)} tuples(R_i)$ (we assume that the values in each $tuples(R_i)$ will be different). Then, any $P \in cover(L)$ will be closed in $\mathcal{R}$.*

Proof. We take any $P \in cover(L)$. It will match an antichain of tuples that will not have any set of tuples that will mix tuples from differents $tuples(P')$ for all $P' \in cover(L)$ and $P \neq P'$ because of Propositions 11 and 12 and the definition of function $tuples(P)$. The antichain will contain a set with $tuples(P)$ because it is matched by P according to definition of $tuples(P)$. For the rest of $tuples(P')$ where $P \neq P'$, there will be no set containing $tuples(P')$, otherwise it would mean that P matches $tuples(P')$ and it would contradict Definition 17. Now, suppose that there is a $P' \neq P$ that matches the same antichain of tuples than P. It means that both would match $tuples(P)$ if they were the only tuples in $\mathcal{R}$ and $tuples(P')$ if they were the only tuples in $\mathcal{R}$, but it would contradict again Definition 17.2. ∎

Theorem 5. *Given $L \in Part(\mathcal{U})$ a lattice that axiomatizes a set of dependencies Σ, then, $\bigcup_{R_i \in cover(L)} tuples(R_i)$ is an Armstrong relation for Σ if for each $tuples(R_i)$, the set of values in all the attributes are different from the values of the rest of $tuples(R_j)$ where $i \neq j$.*

Proof. With the previous propositions, it is easy to see that all the partitions of attributes in $cover(L)$ will be closed. It only remains to prove that the partitions that are the join of some set of partitions will be closed as well. And this is easy to prove since given a partition P that is the join of P^i for a set of partitions $i \in \mathcal{I}$ will match an antichain of tuples such that:

1. There will be no set containing tuples from different $tuples(P^i)$.
2. For all $P^i \preceq P$, there will be the set $tuples(P^i)$.

It will be the only partition of attributes that will match an antichain of tuples that contains sets with $tuples(P^i)$ for all $P^i \preceq P$. ■

References

1. Baixeries J. *A Formal Concept Analysis framework to model functional dependencies.* Mathematical Methods for Learning (2004).
2. 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).
3. Balcázar, J.L., Baixeries J. *Characterizations of multivalued dependencies and related expressions.* Discovery Science 2004.
4. Baixeries J., Balcázar, J.L. *Characterization and Armstrong relations for Degenerated Multivalued Dependencies using Formal Concept Analysis.* The Third International Conference on Formal Concept Analysis 2005.
5. Baixeries J., Balcázar, J.L. *A Lattice Representation of Relations, Multivalued Dependencies and Armstrong Relations.* 13th International Conference on Conceptual Structures (ICCS'05).
6. Baixeries J., Balcázar, J.L. *New Closure Operators and Lattice Representations for Multivalued Dependencies and Related Expressions.* The 3rd International Conference on Concept Lattices and Their Applications (CLA'05).
7. Bastide Y., Taouil R., Pasquier N., Stumme G., Lakhal L. *Mining Frequent Patterns with Counting Inference.* SIGKDD Explorations 2(2): 66-75 (2000).
8. 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.
9. 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.
10. Carpineto C., Romano G. *Inferring Minimal Rule Covers from Relations.* Computational Intelligence, Volume 15, numer 4, 1999, pages 415-441.
11. Davey B.A., Priestley H.A. *Introduction to Lattices and Order.* Second edition. Cambridge University Press, 1990, 2002.
12. Day A. *The lattice theory of functional dependencies and normal decompositions.* International Journal of Algebra Computational 2 (4) (1992) 409-431.
13. 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.
14. Demetrovics J., Hencsey G., Libkin L., Muchnik I. *Normal Form Relation Schemes: A New Characterization.* Acta Cybernetica. 10(3): 141-164 (1992).
15. Demetrovics J., Libkin L., Muchnik I. *Functional Dependencies in Relational Databases: A Lattice Point of View.* Discrete Applied Mathematics 40(2): 155-185 (1992).
16. Fagin R. *Multivalued dependencies and a new normal form for relational databases.* ACM Transactions on Database Systems 2, 3, Sept. 1977, pp. 262-278.

17. 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.
18. Fagin R. *Armstrong databases.* Invited paper, Proc. 7th IBM Symposium on Mathematical Foundations of Computer Science, Kanagawa, Japan, May 1982.
19. Flach P., Savnik I. *Discovery of multivalued dependencies from relations.* Intelligent Data Analysis,volume 4 (3,4): 195–211, November 2000.
20. Ganter, B., Wille R. *Formal Concept Analysis. Mathematical Foundations.* Springer, 1999.
21. Gottlob G., Libkin L. *Investigations on Armstrong relations, dependency inference and excluded functional dependencies.* Acta Cybernetica 9(4): 385-402 (1990).
22. Grätzer, G. *General Lattice Theory.* Birkhäuser Verlag, 2000.
23. 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.
24. Kautz H., Selman B. *Knowledge compilation using Horn approximations.* In Journal of the ACM 43, 2, pages 193-224, 1996.
25. Lopes, S., Petit J-M., Lakhal 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.
26. Mok W. Y., Embley D. W. *On improving Dependency Implication Algorithms.* Information Processing Letters 64 (1997), pages 135 - 141.
27. Pfaltz, J.L. *Transformations of Concept Graphs: An Approach to Empirical Induction.* 2nd International Workshop on Graph Transformation and Visual Modeling Techniques. GTVM 2001, Satellite Workshop of ICALP 2001, Crete, Greece. Pages 320-326. July 2001.
28. Sagiv Y. *An algorithm for inferring multivalued dependencies with an application to propositional logic.* Journal of the ACM, 27(2):250-262, April 1980.
29. Thalheim B. *Entity-Relationship Modelling. Fundamentals of Database Technology.* Springer. Berlin 2000.
30. 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.
31. Ullman J.D. *Principles of Database and Knowledge-Base Systems.* Computer Science Press, Inc. 1988.
32. Valtchev P., Missaoui R. *Building Galois (Concept) Lattices from Parts: Generalizing the Incremental Approach.* Proceedings of the ICCS 2001, LNCS 2120, Springer Verlag, pp. 290-303, Stanford (CA), 2001.
33. Valtchev P., Missaoui R, Lebrun P. *A partition-based approach towards constructing Galois (concept) lattices.* Discrete Mathematics 256(3):801-829, 2002.
34. Zaniolo C., Melkanoff M. A. *On the Design of Relational Database Schemata.* TODS 6(1): 1-47 (1981).

5 Appendices

A The Lattice of Antichains

In this section we prove different properties for the set of antichains of a set, namely, that if a set is finite, then, the set of antichains of this set is a lattice.

Proposition 2. *$\trianglelefteq$ on the set $\wp_S(\wp(Q))$ is an order.*

Proof. $\forall C, C', C'' \in \wp_S(\wp(Q))$:

1. Reflexivity. $C \trianglelefteq C'$: $\forall \mu \in C : \mu \subseteq \mu$ and $\mu \in C$.
2. Antisymmetry. $C \trianglelefteq C'$ and $C' \trianglelefteq C \longrightarrow C = C'$. We take one set $\mu \in C$: since $C \trianglelefteq C'$, then, there is a set $\mu' \in C'$ such that $\mu \subseteq \mu'$ Since $C' \trianglelefteq C$, there is a set $\mu'' \in C$ such that $\mu' \subseteq \mu''$. Since both C, C' are in $\wp_S(\wp(Q))$, then, we have that $\mu = \mu''$ and, hence, $\mu = \mu'$.
3. Transitivity. $C \trianglelefteq C'$ and $C' \trianglelefteq C'' \longrightarrow C \trianglelefteq C''$. We have that $\forall \mu \in C : \exists \mu' \in C' : \mu \subseteq \mu'$ and that $\forall \mu' \in C' : \exists \mu'' \in C'' : \mu' \subseteq \mu''$. Hence, by transitivity of $\subseteq$ we have that $\forall \mu \in C : \exists \mu'' \in C'' : \mu \subseteq \mu''$, it is: $C \trianglelefteq C''$. ∎

Proposition 3. *$simple(C) \trianglelefteq C \trianglelefteq simple(C)$.*

Proof. a. $simple(C) \trianglelefteq C$ holds because $simple(C)$ contains a subset of the sets in C.

b. $C \trianglelefteq simple(C)$: We take any set $\mu \in C$ By Definition 4, $simple(C)$ has the largests sets in C, it is, there will be a set $\mu' \in simple(C)$ such that $\mu \subseteq \mu'$.

∎

Definition 5. *The operations meet and join for the set $\wp_S(\wp(Q))$ are:*

1. $C \wedge C' = simple(C \cap C')$
2. $C \vee C' = simple(C \cup C')$

Proposition 5. *If Q is a finite set, then, $\wp_S(\wp(Q))$ is a complete lattice.*

Proof. We need to prove that the operations meet and join are always defined (they always exist and their result is unique).

1. Meet ($\wedge$): $\forall D \in \wp_S(\wp(Q)) : D \trianglelefteq C$ **and** $D \trianglelefteq C' \longrightarrow D \trianglelefteq C \wedge C'$. We have that $\forall \mu_D \in D : \exists \mu_C \in C : \mu_D \subseteq \mu_C$ and that $\forall \mu_D \in D : \exists \mu'_C \in C' : \mu_D \subseteq \mu'_C$. Then, we have that $\forall \mu_D \in D : \exists \mu_C \in C, \mu'_C \in C' : \mu_D \subseteq \mu_C$ **and** $\mu_D \subseteq \mu'_C$, it is, $\forall \mu_D \in D : \exists \mu_C \in C, \mu'_C \in C' : \mu_D \subseteq \mu_C \cap \mu'_C$. Then, $C \cap C' \trianglerighteq D$. By Proposition 3 $simple(C \cap C') \trianglerighteq C \cap C'$, and by transitivity we have that $simple(C \cap C') = C \vee C' \trianglerighteq D$. ∎
2. Join ($\vee$): $\forall D \in \wp_S(\wp(Q)) : D \trianglerighteq C$ **and** $D \trianglerighteq C' \longrightarrow D \trianglerighteq C \vee C'$. We have that $\forall \mu_C \in C : \exists \mu_D \in D : \mu_C \subseteq \mu_D$ and that $\forall \mu'_C \in C' : \exists \mu_D \in D : \mu'_C \subseteq \mu_D$. Then, we have that $\forall \mu_C \in C, C' : \exists \mu_D \in D : \mu_C \subseteq \mu_D$, it is: $C \cup C' \trianglelefteq D$. By Proposition 3, $simple(C \cup C') \trianglelefteq C \cup C'$ and then, by transitivity, $simple(C \cup C') = C \wedge C' \trianglelefteq D$.

B A Galois Connection Based on the Matching Property

We have two complete lattices P, Q, with an order $\supseteq, \trianglerighteq$ in each of them as well as a binary relation:

Definition 6. *The binary relation **matches** $*$ between P and Q is axiomatized as follows:*

1. $p' \supseteq p$ *and* $p * q \longrightarrow p' * q$.
2. $p * q$ *and* $p' * q \longrightarrow (p \wedge p') * q$.
3. $q \trianglerighteq q'$ *and* $p * q \longrightarrow p * q'$.
4. $p * q$ *and* $p * q' \longrightarrow p * (q \vee q')$.

We now define the following functions for all $p, p' \in P$ and all $q, q' \in Q$:

Definition 7. *The functions* $\varphi : P \mapsto Q$ *and* $\psi : Q \mapsto P$ *are:*

$$\varphi(p) = \bigvee \{q' | p * q'\}$$

$$\psi(q) = \bigwedge \{p' | p' * q\}$$

Theorem 6. *φ, ψ is a Galois connection.*

Proof. We need to prove that $p \supseteq \psi(q) \Leftrightarrow \psi(p) \trianglerighteq q$ ([20]). According to Definition 7, this expression is equivalent to

$$p \supseteq \bigwedge \{p' | p' * q\} \Leftrightarrow \bigvee \{q' | p * q'\} \trianglerighteq q$$

($\Rightarrow$) By Definition 10.2, $\bigwedge\{p'|p' * q\}$ is defined, let it be p'. Then, we have that

$$p \supseteq p' \textbf{ and } p' * q \overset{\text{by Def. 10.1}}{\longrightarrow} p * q \longrightarrow q \in \{q' | p * q'\} \longrightarrow \bigvee \{q' | p * q'\} \trianglerighteq q$$

($\Leftarrow$) By Definition 10.4, $\bigvee\{q'|p * q'\}$ is defined, let it be q'. Then, we have that

$$q' \trianglerighteq q \textbf{ and } p * q' \overset{\text{by Def. 10.3}}{\longrightarrow} p * q \longrightarrow p \in \{p' | p' * q\} \longrightarrow p \supseteq \bigwedge \{p' | p' * q\}$$

■

Note that by Definition 10.1 and 10.2, $\{p'|p' * q\}$ is an ideal, and by 10.3 and 10.4, $\{q'|p * q'\}$ is a filter.