Journal of Universal Computer Science, vol. 15, no. 1 (2009), 112-156
submitted: 28/1/08, accepted: 3/5/08, appeared: 1/1/09 © *J.UCS*

Weak Functional Dependencies: Full Propositional Expressiveness for the Database Practitioner

Sven Hartmann
(Department of Informatics, Clausthal University of Technology, Germany
sven.hartmann@tu-clausthal.de)

Sebastian Link
(School of Information Management, Victoria University of Wellington
New Zealand
sebastian.link@vuw.ac.nz)

Abstract: We study inference systems of weak functional dependencies in relational and complex-value databases. Functional dependencies form a very common class of database constraints. Designers and administrators proficiently utilise them in everyday database practice. Functional dependencies correspond to the linear-time decidable fragment of Horn clauses in propositional logic. Weak functional dependencies take advantage of arbitrary clauses, and therefore represent full propositional reasoning about data in databases. Moreover, they can be specified in a way that is very similar to functional dependencies.

In relational databases the class of weak functional dependencies is finitely axiomatisable and the associated implication problem is *coNP*-complete in general. Our first main result extends this axiomatisation to databases in which complex elements can be derived from atomic ones by finitely many nestings of record, list and disjoint union constructors. In particular, we construct two nested tuples that can serve as a counterexample relation for the implication of weak functional dependencies. We further apply this construction to show an equivalence to truth assignments that serve as counterexamples for the implication of propositional clauses. Hence, we characterise the implication of weak functional dependencies in complex-value databases in completely logical terms. Consequently, state-of-the-art SAT solvers can be applied to reason about weak functional dependencies in relational and complex-value databases.

Key Words: Relational database, Complex-value database, Weak functional dependency, Axiomatisation, Propositional logic
Category: F.4, H.2

1 Introduction

The semantics of databases is usually captured at the time of designing their database schemata. While a schema itself imposes structural constraints the specification of additional constraints can further restrict the number of possible future database instances to those which are considered meaningful to the application at hand. A common approach formally declares so-called dependencies that form a common class of constraints. Intuitively, dependencies mean

that the occurrence of data satisfying certain properties enforce the existence or properties of other data. In this sense, the latter data entries are dependent on the former ones. While this rough intuition is useful for verbal communication the exact formal meaning and a correct understanding by users and administrators requires a strictly formal treatment of mathematical rigour. For relational databases almost 100 different classes of dependencies are known [Fagin and Vardi, 1986; Thalheim, 1991].

In database practice only a few of these classes are actually utilised. Functional dependencies form a prime example of such a class. Informally, a relation exhibits a functional dependency if whenever two table rows agree on a set of columns, then they also agree on another set of columns. Possible reasons for the frequent utilisation of functional dependencies are their significance for capturing essential requirements, their simplicity, the proficiency with which data administrators can employ them, and their robustness in terms of maintenance and understanding. For instance, the interaction of functional dependencies corresponds exactly to the interaction of Horn clauses in propositional logic. This limited expressiveness, however, explains why functional dependencies cannot capture several properties that are desirable in many database applications. In reality, administrators find it difficult to specify more expressive dependencies. This bottleneck constricts further when the structure of the database schema cannot be represented as a table but by means of more sophisticated schemata such as nested schemata [Levene, 1992] or document type definitions [Bray et al., 2006].

We propose to use *weak functional dependencies* as an expressive class of database dependencies that can be specified in a way similar to functional dependencies. Informally, a relation exhibits a weak functional dependency if whenever two table rows agree on a set of columns, then they also agree on at least one column out of a set of columns. This appears to be just a minor modification of the semantics for functional dependencies, but yields the full expressiveness of propositional logic. Indeed, as we will show, the specification of a finite set of weak functional dependencies corresponds to a formula of propositional logical in conjunctive normal form (with each clause corresponding to a weak functional dependency in that set).

Example 1. Consider the relation schema DEPARTMENT with attributes ID, Staff, Level, Semester, Project, Teaching which keeps record of staff members' teaching duties and involvement in projects. The department has the policy that a professor teaches one course per semester and is involved in at least one project. Other staff members must teach at least one course and participate in one project. An instance of DEPARTMENT is shown in Table 1. According to the department's policy, whenever two tuples coincide on Staff and Semester, then these two tuples coincide on Teaching (in case of a professor) or they coincide

ID	Staff	Level	Semester	Project	Teaching
123	John Do	Professor	2005-01	Weak FDs	Relational Databases
123	John Do	Professor	2005-01	Strong FDs	Relational Databases
123	John Do	Professor	2005-02	Weak FDs	XML
123	John Do	Professor	2005-02	Data Exchange	XML
123	John Do	Professor	2005-02	Query Answering	XML
234	Frank Gorge	Professor	2005-01	Strong FDs	Temporal Databases
234	Frank Gorge	Professor	2005-01	XQuery	Temporal Databases
345	Bo Candle	Lecturer	2005-01	XPath	Relational Databases
345	Bo Candle	Lecturer	2005-01	XPath	Temporal Databases
345	Bo Candle	Lecturer	2005-01	XPath	Spatial Databases
345	Bo Candle	Lecturer	2005-02	Data Exchange	Database Security

Table 1: A relation over DEPARTMENT

on Project (in case of a staff member who is not a professor). Therefore, every legal instance over DEPARTMENT is to satisfy the weak functional dependency

$$\{\text{Staff, Semester}\} \rightarrow_w \{\text{Project, Teaching}\}.$$

Furthermore, the staff member is functionally determined by its ID, and every staff member is associated with one level. We can express these functional dependencies as weak functional dependencies

$$\text{ID} \rightarrow_w \text{Staff} \qquad \text{and} \qquad \text{Staff} \rightarrow_w \text{Level}\ .$$

We choose {ID, Semester, Teaching, Project} as primary key for DEPARTMENT, i.e., every two distinct tuples disagree on at least one attribute in {ID, Semester, Teaching, Project}. □

Example 1 shows that weak functional dependencies can be used to express keys and functional dependencies, but also many other properties.

Previous Work. Weak functional dependencies have been introduced and studied for the relational model of data in [Demetrovics and Gyepesi, 1981]. Recently, there has been interest in extending dependencies to databases that can handle data elements that are derived from atomic ones by finite recursive nestings of certain type constructors [Hartmann and Link, 2008; Hartmann et al., 2004; Hartmann et al., 2005; Hartmann et al., 2006a; Hartmann et al., 2006b; Sali and Schewe, 2006; Schewe, 2005]. Weak functional dependencies are treated as a subclass of disjunctive functional dependencies in [Hartmann et al., 2004; Hartmann et al., 2005]. The underlying framework is based on non-trivial restructuring rules that result in a very sophisticated axiomatisation.

Contributions. In this article we make the following contributions:

1. We prove the soundness and completeness for a set of inference rules for the implication of weak functional dependencies in relational databases.

2. Based on type constructors for records, lists and disjoint unions we introduce nested database schemata that capture the structural information on complex-value databases. These schemata carry the structure of a Brouwerian algebra which deviates from previously studied algebras in the subtyping rules for deriving subschemata. The new Brouwerian algebra carries additional domain information which is especially suitable in the presence of disjoint union types and weak functional dependencies. In sharp contrast to the work in [Hartmann et al., 2004; Hartmann et al., 2005] we do not consider any restructuring rules. This results in an orthogonal and easily comprehensible theory.

3. We extend the definition of weak functional dependencies from relational to complex-value databases. In particular, it is sufficient to focus on the join-irreducible subschemata of a nested attribute. This is different to the situation when set- or multiset constructor are present [Hartmann et al., 2006a].

4. We extend the axiomatisation for the implication of weak functional dependencies to complex-value databases. The major difficulty is the construction of two nested data elements that can serve as a counterexample for an instance of the implication problem. The combination of list and disjoint union constructor imposes additional challenges on the construction.

5. We characterise the implication of weak functional dependencies in complex-value databases in completely logical terms. In fact, truth assignments that serve as counterexamples for the implication of propositional formulae have a one-to-one correspondence to the two-element relations that serve as counterexamples for the implication of weak functional dependencies. The correspondence shows that weak functional dependencies form a very robust class of dependencies, and that well-studied state-of-the-art reasoning tools can be applied to reason about them. On the one hand, the data administrator can take advantage of her proficiency in specifying functional dependencies to define weak functional dependencies. On the other hand, the full propositional expressive power is at her disposal. Indeed, functional dependencies correspond to propositional Horn clauses, weak functional dependencies correspond to propositional clauses in general.

Organisation. We study weak functional dependencies for relational databases in Section 2. In Section 3 we introduce the complex-value data model and establish an axiomatisation for the implication of weak functional dependencies in that framework. The correspondence between weak functional dependencies and

propositional formulae is established in Section 4. We conclude in Section 5 and briefly comment on future work.

2 Weak Functional Dependencies in Relational Databases

Let $\mathfrak{A} = \{A_1, A_2, \ldots\}$ be a (countably) infinite set of distinct symbols called *attributes*. A *relation schema* is a non-empty finite set $R = \{A_1, \ldots, A_k\}$ of attributes. Each attribute A_i of a relation schema has a non-empty, possibly infinite, domain $dom(A_i)$. If X and Y are sets of attributes, then we may write XY for $X \cup Y$. If $X = \{A_1, \ldots, A_m\}$, then we may write $A_1 \cdots A_m$ for X. In particular, we may write simply A to represent the singleton $\{A\}$. A *tuple* over $R = \{A_1, \ldots, A_k\}$ (R-tuple or simply tuple, if R is understood) is a function $t : R \to \bigcup_{i=1}^{n} dom(A_i)$ with $t(A_i) \in dom(A_i)$ for $i = 1, \ldots, k$. For $X \subseteq R$ let $t[X]$ denote the restriction of the tuple t over R to X, and $dom(X) = \prod_{A \in X} dom(A)$ the Cartesian product of the domains of attributes in X. A *relation* r over R (or R-relation for short) is a finite set of tuples over R. Let $r[X] = \{t[X] \mid t \in r\}$ denote the *projection* of the relation r over R on $X \subseteq R$.

2.1 Weak FDs and Clauses in Boolean Propositional Logic

Originally, weak functional dependencies have been introduced for relational databases in [Demetrovics and Gyepesi, 1981] as generalisations of functional dependencies.

A *weak functional dependency* (wFD) on the relation schema R is an expression $X \to_w Y$ where $X, Y \subseteq R$. A relation r over R *satisfies* the wFD $X \to_w Y$, denoted by $\models_r X \to_w Y$, if and only if for every pair of distinct tuples in r that agree on each of the attributes in X, they also agree on at least one attribute in Y. That is, $\models_r X \to Y$ if and only for every distinct $t_1, t_2 \in r$ such that for all $A \in X$, $t_1[A] = t_2[A]$ holds, there is some $B \in Y$ such that $t_1[B] = t_2[B]$ holds.

We will assume from now on that the domain $dom(A)$ of every attribute A contains at least two distinct elements. In practice, an attribute with a singleton as its domain is not really required as the value of this attribute will be fixed. Hence, it may therefore be omitted completely.

For a set $\Sigma \cup \{\varphi\}$ of wFDs on a relation schema R we say that Σ *implies* φ if and only if every R-relation $r \subseteq dom(R)$ that satisfies all the wFDs in Σ also satisfies φ. Hence, Σ *implies* φ precisely when there is no counterexample relation r that satisfies all the wFDs in Σ but violates φ. The *implication problem* for weak functional dependencies is to decide whether for an arbitrary relation schema R and an arbitrary set $\Sigma \cup \{\varphi\}$ of wFDs on R the wFD φ is implied by Σ.

For the design of a relational database schema dependencies are normally specified as semantic constraints on the relations which are intended to be instances of the schema. The design process requires the data administrator to determine further dependencies which are implied by the given ones. In order to determine the logical consequences of a set of dependencies one can use inference rules of the form

$$\frac{\text{premise}}{\text{conclusion}}$$

and inference rules without a premise are called *axioms*.

Let $\Sigma \cup \{\varphi\}$ be a set of wFDs on the relation schema R. We use $\mathfrak{S}$ to denote a set of inference rules. Let $\Sigma \vdash_{\mathfrak{S}} \varphi$ denote the inference of φ from Σ with respect to $\mathfrak{S}$. Let $\Sigma^+_{\mathfrak{S}} = \{\varphi \mid \Sigma \vdash_{\mathfrak{S}} \varphi\}$ denote the *syntactic hull* of Σ under inference using only rules from $\mathfrak{S}$. An inference rule is called *sound* if the set of dependencies in the premise of the rule implies the dependency in the conclusion. The set $\mathfrak{S}$ is *sound* for the implication of wFDs if and only if for every relation schema R and for every set Σ of wFDs on R we have $\Sigma^+_{\mathfrak{S}} \subseteq \Sigma^*_R = \{\varphi \mid \Sigma \text{ implies } \varphi\}$. The set $\mathfrak{S}$ is called *complete* for the implication of wFDs if and only if for every relation schema R and for every set Σ of wFDs on R we have $\Sigma^*_R \subseteq \Sigma^+_{\mathfrak{S}}$.

In what follows we assume familiarity with basic notions from propositional logic [Enderton, 2001]. Weak functional dependencies form a subclass of Boolean dependencies [Sagiv et al., 1981; Sagiv et al., 1987]. In fact, every wFD $A_1 \dots A_n \rightarrow_w B_1 \dots B_m$, denoted by φ, represents the clause $\neg A_1 \vee \dots \vee \neg A_n \vee B_1 \vee \dots \vee B_m$, denoted by φ', where $A_1, \dots, B_m$ are interpreted as propositional variables. With this mapping $(\cdot)'$ of wFDs to propositional clauses in mind, the wFD φ is implied by the set Σ of wFDs if and only if the corresponding propositional clause φ' is logically implied by the set $\Sigma' = \{\sigma' \mid \sigma \in \Sigma\} \cup \{(R \rightarrow_w \emptyset)'\}$ of corresponding propositional clauses. The proof utilises a strong correspondence between two-tuple counterexample relations r for the implication of φ by Σ and counterexample truth assignments θ_r for the logical implication of φ' by $\Sigma' \cup \{(R \rightarrow_w \emptyset)'\}$. More precisely, the variable A is assigned the truth value *true* under θ_r if and only if the two tuples of r agree on the attribute A [Sagiv et al., 1981]. In particular, the propositional clause $(R \rightarrow_w \emptyset)'$ corresponds to the set semantics of database relations, i.e., every pair of distinct tuples must disagree on some attribute.

Recall that the *satisfiability problem*, i.e. the problem of deciding whether an arbitrary finite set of propositional clauses has a model, is *NP*-complete [Cook, 1971; Levine, 1973].

Theorem 1. *The implication problem of weak functional dependencies is* coNP-*complete.* □

If we restrict our attention to wFDs in which the right-hand sides contain at most one attribute, then we have an equivalence to propositional Horn clauses, i.e., to functional dependencies in relational databases [Fagin, 1977a]. The following corollary follows from the linear-time decidability of propositional Horn clauses [Dowling and Gallier, 1984].

Corollary 2. *The implication problem of wFDs with at most one attribute in the right-hand side can be decided in time linear in the total number of attributes occurring in the input instance.* □

2.2 Weak Functional Dependencies and Keys

The relation r satisfies the wFD $X \rightarrow_w \emptyset$ if and only if the projections $t_1[X]$ and $t_2[X]$ deviate from one another for every pair t_1, t_2 of distinct tuples in r. In other words, the attribute set X forms a key. That is, no relation over R contains two distinct tuples with the same projection on X. A wFD $X \rightarrow_w \emptyset$ on R is called a key dependency on R. Given a set Σ of key dependencies on R, what are the key dependencies implied by Σ? The following two inference rules

$$\frac{}{R \rightarrow_w \emptyset} \qquad \frac{X \rightarrow_w \emptyset}{XY \rightarrow_w \emptyset}$$

are sound and complete for the implication of key dependencies. The soundness of the rules is not difficult to see. For the completeness assume that $X \rightarrow_w \emptyset \notin \Sigma^+$ for some $X \subseteq R$. Then $R - X \neq \emptyset$ since otherwise $X \rightarrow_w \emptyset \in \Sigma^+$ by application of the first inference rule. Define two tuples $t_1, t_2 \in dom(R)$ such that $t_1[A] = t_2[A]$ if and only if $A \in X$. Note that this defines indeed a relation with two distinct tuples. It follows that $r = \{t_1, t_2\}$ does not satisfy $X \rightarrow_w \emptyset$. It remains to show that r satisfies every key dependency in Σ. Let $U \rightarrow_w \emptyset \in \Sigma$. If $U \subseteq X$, then $X \rightarrow_w \emptyset \in \Sigma^+$ according to the second inference rule. Consequently, there is some attribute $A \in U - X$. This means that $t_1[A] \neq t_2[A]$ and therefore $\models_r U \rightarrow_w \emptyset$. We conclude that $X \rightarrow_w \emptyset \notin \Sigma^*$, i.e., the completeness follows.

Example 2. Consider again Example 1. The weak functional dependency $X \rightarrow_w \emptyset$ expresses that $X = \{$ID,Semester,Teaching,Project$\}$ is a key for DEPARTMENT, i.e., every pair of distinct tuples must deviate on at least one attribute in X. □

2.3 An Axiomatisation of Weak FDs

Weak functional dependencies have been axiomatised before [Demetrovics and Gyepesi, 1981; Thalheim, 1991]. However, we provide a new proof for the completeness of the set $\mathfrak{R}$ of inference rules from Table 2. Instead of using combinatorial arguments to show the completeness [Demetrovics and Gyepesi, 1981;

$\dfrac{}{X \to_w Y}\ X \cap Y \neq \emptyset$ (Axiom 1)	$\dfrac{}{R \to_w \emptyset}$ (Axiom 2)	$\dfrac{X \to_w Y}{X \to_w Z}\ Y \subseteq Z$ (Rule 1)
$\dfrac{X \to_w Y}{Z \to_w Y}\ X \subseteq Z$ (Rule 2)	$\dfrac{\{XV \to_w Y(U-V)\}_{V \subseteq U}}{X \to_w Y}$ (Rule 3)	

Table 2: An Axiomatisation of wFDs in relational databases

Thalheim, 1991], for each wFD φ that cannot be derived by a set Σ of wFDs we construct a two tuple relation that forms a counterexample for the implication of φ by Σ. The proof will also serve as a good preparation for the more technical proof that is required in the framework of complex-value databases.

Theorem 3. *The set $\mathfrak{R}$ of inference rules from Table 2 is sound and complete for the implication of wFDs.*

Proof. We show the soundness of the inference rules first. Consider *Axiom 1*. If $X \cap Y \neq \emptyset$, then there is some $A \in X \cap Y$. For an arbitrary R-relation r and any two distinct tuples $t_1, t_2 \in r$ with $t_1[X] = t_2[X]$ we have $t_1[A] = t_2[A]$ as $A \in X$, in particular. Consequently, $\models_r X \to_w Y$.

Consider *Axiom 2*. For an arbitrary R-relation r any pair t_1, t_2 of distinct tuples in r there is at least one attribute $A \in R$ with $t_1[A] \neq t_2[A]$. That is, $\models_r R \to_w \emptyset$.

Consider *Rule 1*. Let r be some arbitrary R-relation that satisfies the wFD $X \to_w Y$. For any two distinct $t_1, t_2 \in r$ with $t_1[X] = t_2[X]$ there is then some $A \in Y$ with $t_1[A] = t_2[A]$. Consequently, there is also some $A \in Z$ with $t_1[A] = t_2[A]$ since $Y \subseteq Z$. This shows that r also satisfies the wFD $X \to_w Z$.

Consider *Rule 2*. Let r be some arbitrary R-relation that satisfies the wFD $X \to_w Y$. For any two distinct $t_1, t_2 \in r$ with $t_1[Z] = t_2[Z]$ it follows that $t_1[X] = t_2[X]$ as $X \subseteq Z$ holds. Consequently, there is some $A \in Y$ with $t_1[A] = t_2[A]$ since $\models_r X \to_w Y$. This shows that r also satisfies the wFD $Z \to_w Y$.

Consider *Rule 3*. Let r be some arbitrary R-relation that satisfies all wFDs $XV \to_w Y(U - V)$ for all subsets $V \subseteq U$ of some $U \subseteq R$. Let $t_1, t_2 \in r$ be distinct with $t_1[X] = t_2[X]$. Since r satisfies the wFD $X \to_w YU$ there is some $A \in YU$ with $t_1[A] = t_2[A]$. If $A \in Y$, then there is nothing more to show. Otherwise, $A \in U$. Since r satisfies the wFD $XA \to_w Y(U - A)$ there is then some $B \in Y(U - A)$ with $t_1[B] = t_2[B]$. If $B \in Y$, then there is nothing more to show. Otherwise, we continue this line of reasoning until we have found that

$t_1[XU] = t_2[XU]$. Since r also satisfies $XU \rightarrow_w Y$ there is some $D \in Y$ with $t_1[D] = t_2[D]$. Consequently, r satisfies also the wFD $X \rightarrow_w Y$.

The soundness of the inference system follows from the soundness of the inference rules by a simple induction on the length of an inference.

We will now show the completeness of the rules. Let $\Sigma \cup \{X \rightarrow_w Y\}$ be a set of wFDs on the relation schema R. Suppose that $X \rightarrow_w Y \notin \Sigma^+$. We need to show that $X \rightarrow_w Y \notin \Sigma^*$. We will construct a two-tuple relation r that satisfies all wFDs in Σ, but violates $X \rightarrow_w Y$.

First, it follows that $X \cap Y = \emptyset$ since, otherwise, $X \rightarrow Y \in \Sigma^+$ by Axiom 1. It also follows that $R - X \neq \emptyset$ since, otherwise, $X = R$ and $Y = \emptyset$ and, therefore, $X \rightarrow Y \in \Sigma^+$ by Axiom 2.

Define $U := R - (XY)$. Then there is some $V \subseteq U$ such that $XV \rightarrow_w Y(U - V) \notin \Sigma^+$. Otherwise, $X \rightarrow_w Y \in \Sigma^+$ by Rule 3. We conclude that $Y(U - V) \neq \emptyset$ since, otherwise, $Y = \emptyset$, $V = U$ and $XV = R$ which implies $XV \rightarrow Y(U - V) \in \Sigma^+$ by Axiom 2. Note that the sets $X, Y, V, U - V$ form a partition of R. Define tuples $t_1, t_2 \in dom(R)$ such that

$$t_1[A] = t_2[A] \qquad \text{if and only if} \qquad A \in XV.$$

The tuples t_1 and t_2 are indeed distinct since $Y(U - V) \subseteq R - (XV)$ and $Y(U - V) \neq \emptyset$.

It follows that $t_1[X] = t_2[X]$ since $X \subseteq XV$. However, $t_1[A] \neq t_2[A]$ for all $A \in Y$ since $Y \subseteq R - (XV)$. Therefore, r does not satisfy $X \rightarrow_w Y$.

We will show now that r does satisfy all wFDs in Σ. Therefore, let $W \rightarrow_w Z \in \Sigma$ such that $t_1[W] = t_2[W]$ holds. Otherwise, $W \rightarrow_w Z$ is satisfied by r. We show that there is some attribute $A \in Z$ on which t_1 and t_2 agree.

Since $t_1[W] = t_2[W]$ we conclude that $W \subseteq XV$ by the construction of r. Since $W \rightarrow_w Z \in \Sigma$ and $W \subseteq XV$ it follows that $XV \rightarrow_w Z \in \Sigma^+$ by Rule 2. Assume that $Z \subseteq Y(U - V)$. Then Rule 1 implies that $XV \rightarrow_w Y(U - V) \in \Sigma^+$, a contradiction to our previous choice of V. This means $Z \not\subseteq Y(U - V)$. However, $Y(U - V) = R - (XV)$ by definition of U. Consequently, $Z \not\subseteq R - (XV)$ which means that $Z \cap (XV) \neq \emptyset$. Consequently, there is some attribute $A \in Z \cap (XV)$, and $t_1[A] = t_2[A]$ by construction of r. We conclude that r satisfies $W \rightarrow_w Z$.

We have shown that r satisfies every wFD in Σ which implies that r satisfies every wFD in Σ^*. Since r does not satisfy $X \rightarrow_w Y$ it follows that $X \rightarrow_w Y \notin \Sigma^*$. This shows the completeness. □

The following example demonstrates the applicability of the inference rules in $\mathfrak{R}$.

Example 3. Consider the relation schema DEPARTMENT from Example 1. We may apply (Rule 1) to the wFD ID $\rightarrow_w$ Staff in order to infer ID $\rightarrow_w$ {Staff, Level}. Furthermore, we apply (Rule 2) to the wFD Staff $\rightarrow_w$ Level to infer {ID, Staff} $\rightarrow_w$ Level. If U = {Staff}, then we may apply (Rule 3) to the wFDs

$$\{\text{ID, Staff}\} \rightarrow_w \text{Level} \qquad \text{and} \qquad \text{ID} \rightarrow_w \{\text{Staff, Level}\}$$

and infer the wFD ID $\rightarrow_w$ Level. This inference corresponds to an application of the transitivity rule for functional dependencies.

We apply (Rule 1) and (Rule 2) to infer {ID, Semester} $\rightarrow_w$ {Name, Teaching, Project} from ID $\rightarrow_w$ Name. Moreover, we apply (Rule 2) to infer {ID, Name, Semester} $\rightarrow_w$ {Teaching, Project} from {Name, Semester} $\rightarrow_w$ {Teaching, Project}. If $U = \{\text{Name}\}$, then we can apply (Rule 3) to infer

$$\{\text{ID, Semester}\} \rightarrow_w \{\text{Teaching, Project}\}. \qquad \square$$

3 Weak FDs in the Presence of Complex-values

Complex-value data models have been proposed to overcome severe limitations of the relational model of data when designing many practical database applications [Abiteboul et al., 1995].

In this section we will study weak functional dependencies in data models that can deal with complex data. In order assess the impact of complex type constructors on the theory of wFDs we will not restrict our attention to any specific data model. Instead, we will use nested attributes that can be generated from flat attributes by an arbitrary but finite number of recursive applications of record, list and disjoint union operators. Subsequently, our findings may be applied and adjusted to other, more specific data models. For instance, concatenation, Kleene closure and optionality in document type definitions for XML [Bray et al., 2006] represent applications of the record, list and union constructor, respectively.

3.1 Nested Attributes

We start with the definition of flat attributes and values for them. A *universe* is a finite set $\mathcal{U}$ together with domains (i.e. sets of values) $dom(A)$ for all $A \in \mathcal{U}$. The elements of $\mathcal{U}$ are called *flat attributes*. Flat attributes will be denoted by upper-case characters from the start of the alphabet such as A, B, C etc.

In the following we will use a finite set $\mathcal{L}$ of labels that is disjoint from $\mathcal{U}$. These labels are used to improve the readability of nested database schemata. This situation extends the relational case where the symbol R may denote the relation schema $\{A_1, \ldots, A_k\}$. For each label $L \in \mathcal{L}$ there is a distinguished symbol $\lambda_L \notin \mathcal{L} \cup \mathcal{U}$. Moreover, for each flat attribute $A \in \mathcal{U}$ there is a distinguished symbol $\lambda_A \notin \mathcal{L} \cup \mathcal{U}$.

Database schemata in our data model will be given in form of nested attributes. Let $\mathcal{U}$ be a universe and $\mathcal{L}$ a set of labels. The set $\mathcal{N}(\mathcal{U}, \mathcal{L})$ of *nested attributes over* $\mathcal{U}$ *and* $\mathcal{L}$ is the smallest set satisfying the following conditions:

- $\lambda_L \in \mathcal{N}(\mathcal{U}, \mathcal{L})$ for every $L \in \mathcal{L}$,
- $\lambda_A \in \mathcal{N}(\mathcal{U}, \mathcal{L})$ for every $A \in \mathcal{U}$,
- $\mathcal{U} \subseteq \mathcal{N}(\mathcal{U}, \mathcal{L})$,
- for $L \in \mathcal{L}$ and $N_1, \ldots, N_k \in \mathcal{N}(\mathcal{U}, \mathcal{L})$ with $k \geq 1$ we have $L(N_1, \ldots, N_k) \in \mathcal{N}(\mathcal{U}, \mathcal{L})$,
- for $L \in \mathcal{L}$ and $N \in \mathcal{N}(\mathcal{U}, \mathcal{L})$ we have $L[N] \in \mathcal{N}(\mathcal{U}, \mathcal{L})$
- for $L \in \mathcal{L}$ and $N_1, \ldots, N_k \in \mathcal{N}(\mathcal{U}, \mathcal{L})$ with $k \geq 2$ we have $L(N_1 \oplus \cdots \oplus N_k) \in \mathcal{N}(\mathcal{U}, \mathcal{L})$.

We call λ_L the *null attribute of label* L, λ_A the *null attribute of flat attribute* A, $L(N_1, \ldots, N_k)$ *record-valued attribute*, $L[N]$ *list-valued attribute* and $L(N_1 \oplus \cdots \oplus N_k)$ *union-valued attribute*. From now on, we assume that a set $\mathcal{U}$ of flat attribute names, and a set $\mathcal{L}$ of labels is fixed, and write $\mathcal{N}$ instead of $\mathcal{N}(\mathcal{U}, \mathcal{L})$. Null attributes are distinguished attributes whose domain is a single null value which indicates that some information exists but has currently been masked out. Notice that in previous approaches towards defining nested attributes [Hartmann et al., 2004; Hartmann et al., 2005; Hartmann and Link, 2008; Hartmann et al., 2006a; Sali and Schewe, 2006; Schewe, 2005] different null attributes were not considered. Notice that the different null attributes can provide useful domain information.

Example 4. Consider the record-valued attribute $R(A_1, \ldots, A_k)$. This is just a different notation of a relation schema $R = \{A_1, \ldots, A_k\}$. Similarly, the record-valued attribute $R(A_1, A_2, \lambda_{A_3}, \ldots, A_k)$ may represent the subset $\{A_1, A_2\}$ of R.

Consider the nested attribute $N = K(L[M(A, B)] \oplus C)$. It models a disjoint union of lists of pairs of elements from A and B, and elements of C. The nested attribute $K(L[M(\lambda_A, B)] \oplus C)$ models elements of the same structure as before, but the information on the elements of A has been masked out. Finally, the nested attribute $K(L[\lambda_M] \oplus \lambda_C)$ models a disjoint union of lists of elements that have a structure modelled by M and elements from the domain of C. □

We can extend the mapping *dom* from flat attributes to nested attributes, i.e., we define a set $dom(N)$ of values for every nested attribute $N \in \mathcal{N}$. For a nested attribute $N \in \mathcal{N}$ we define the *domain* $dom(N)$ as follows:

- $dom(\lambda_L) = \{ok_L\}$,
- $dom(\lambda_A) = \{ok_A\}$,
- $dom(A)$ for $A \in \mathcal{U}$ as before,

- $dom(L(N_1, \ldots, N_k)) = \prod_{i=1}^{k} dom(N_i)$, i.e., the set of all k-tuples $(v_1, \ldots, v_k)$ with $v_i \in dom(N_i)$ for all $i = 1, \ldots, k$,
- $dom(L[N]) = dom(N)^*$, i.e., the set of all finite lists $[v_1, \ldots, v_n]$ with elements in $dom(N)$,
- $dom(L(N_1 \oplus \cdots \oplus N_k)) = \bigcup_{i=1}^{k} dom(N_i)$, i.e., the *disjoint* union of the domains for $N_1, \ldots, N_k$.

We assume for the remainder of the paper that domains $dom(A)$ of flat attributes A contain at least two different elements. The empty list is denoted by $[\,]$.

Example 5. Consider the nested attribute $N = K(L[M(\lambda_A, B)] \oplus C)$. If b, b' denote elements of $dom(B)$ and c denotes an element of $dom(C)$, then

$$[(ok_A, b), (ok_A, b'), (ok_A, b')]$$

and c denote two elements of $dom(N)$. Consider now the nested attribute $N' = K(L[\lambda_M] \oplus \lambda_C)$. Then $[ok_L, ok_L, ok_L]$ and ok_C denote elements of $dom(N')$. □

3.2 Subattributes

The replacement of attributes by their corresponding null attributes within a nested attribute decreases the amount of information that is being modelled. This observation allows us to introduce an order between nested attributes.

The *subattribute relation* $\leq$ on the set of nested attributes $\mathcal{N}$ over $\mathcal{U}$ and $\mathcal{L}$ is defined by the following rules, and the following rules only:

- $N \leq N$,
- $\lambda_A \leq A$ for all flat attributes $A \in \mathcal{U}$,
- $\lambda_L \leq L[M]$ and $\lambda_L \leq L(N_1 \oplus \cdots \oplus N_k)$ for nested attributes $M, N_1, \ldots, N_k \in \mathcal{N}$,
- $L(N_1, \ldots, N_k) \leq L(M_1, \ldots, M_k)$ whenever $N_i \leq M_i$ for all $i = 1, \ldots, k$,
- $L[N] \leq L[M]$ whenever $N \leq M$, and
- $L(N_1 \oplus \cdots \oplus N_k) \leq L(M_1 \oplus \ldots \oplus M_k)$ whenever $N_i \leq M_i$ for all $i = 1, \ldots, k$.

For N, M we say that M is a *subattribute* of N if and only if $M \leq N$ holds. We write $M \not\leq N$ if M is not a subattribute of N, and $M < N$ in case $M \leq N$ and $M \neq N$.

Lemma 4. *The subattribute relation is a partial order on nested attributes.* □

The subattribute relationship between nested attributes generalises the inclusion relationship between sets of attributes in the relational data model.

Example 6. Consider again the relation schema

$$\text{DEPARTMENT}=\{\text{ID, Staff, Level, Semester, Project, Teaching}\}$$

from Example 1. Alternatively, we may view the name DEPARTMENT as a label and the relation schema as the nested attribute

$$N = \text{DEPARTMENT}(\text{ID}, \text{Staff}, \text{Level}, \text{Semester}, \text{Project}, \text{Teaching}).$$

For instance, the subset {Staff,Semester,Project} of the relation schema DEPARTMENT becomes the subattribute

$$\text{DEPARTMENT}(\lambda_{\text{ID}}, \text{Staff}, \lambda_{\text{Level}}, \text{Semester}, \text{Project}, \lambda_{\text{Teaching}})$$

of the nested attribute N. In fact, the powerset algebra induced by the subset relationship on DEPARTMENT is isomorphic to the Boolean algebra induced by the subattribute relationship on N, see Theorem 5.

Consider now the nested attribute $N = K(L[M(A, B)] \oplus C)$. The subattributes

$$K(L[M(\lambda_A, B)] \oplus C), \quad K(L[M(\lambda_A, B)] \oplus \lambda_C), \quad K(L[\lambda_M] \oplus \lambda_C)$$

form a $\leq$-descending chain of subattributes on N. □

Informally, M is a subattribute of N if and only if M comprises at most as much information as N does. The informal description of the subattribute relation is formally documented by the existence of a projection function $\pi_M^N : dom(N) \to dom(M)$ in case $M \leq N$ holds. For $M \leq N$ the *projection function* $\pi_M^N : dom(N) \to dom(M)$ is defined as follows:

- if $N = M$, then $\pi_M^N = id_{dom(N)}$ is the identity on $dom(N)$,
- if $N = A \in \mathcal{U}$ and $M = \lambda_A$, then $\pi_M^N : dom(A) \to \{ok_A\}$ is the constant function that maps every $a \in dom(A)$ to ok_A,
- if $N = L[N']$ or $N = L(N_1 \oplus \cdots \oplus N_k)$ and $M = \lambda_L$, then $\pi_M^N : dom(N) \to \{ok_L\}$ is the constant function that maps every $v \in dom(N)$ to ok_L,
- if $N = L(N_1, \ldots, N_k)$ and $M = L(M_1, \ldots, M_k)$, then $\pi_M^N = \pi_{M_1}^{N_1} \times \cdots \times \pi_{M_k}^{N_k}$ which maps every tuple $(v_1, \ldots, v_k) \in dom(N)$ to $(\pi_{M_1}^{N_1}(v_1), \ldots, \pi_{M_k}^{N_k}(v_k)) \in dom(M)$,
- if $N = L[N']$ and $M = L[M']$, then $\pi_M^N : dom(N) \to dom(M)$ maps every list $[v_1, \ldots, v_n] \in dom(N)$ to the list $[\pi_{M'}^{N'}(v_1), \ldots, \pi_{M'}^{N'}(v_n)] \in dom(M)$, and

– if $N = L(N_1 \oplus \cdots \oplus N_k)$ and $M = L(M_1 \oplus \ldots \oplus M_k)$, then $\pi^N_M : dom(N) \to dom(M)$ maps every $v \in dom(N_i)$ to $\pi^{N_i}_{M_i}(v) \in dom(M)$.

The domain $dom(N)$ of the union-valued attribute $N = L(N_1 \oplus \cdots \oplus N_k)$ is the disjoint union of the domains $dom(N_i)$. That is, every $v \in dom(N)$ belongs to exactly one $dom(N_i)$.

Example 7. Consider again the nested attribute $N = K(L[M(A, B)] \oplus C)$, its subattribute $X = K(L[M(\lambda_A, B)] \oplus \lambda_C)$ and the tuples $t_1 = [(a', b), (a, b')] \in dom(L[M(A, B)])$ and $t_2 = c \in dom(C)$. The projections of t_1 and t_2 on X are

$$\pi^N_X(t_1) = [(ok_A, b), (ok_A, b')] \quad \text{and} \quad \pi^N_X(t_2) = ok_C, \text{respectively.}$$

The projections of t_1 and t_2 on $K(\lambda_L \oplus \lambda_C)$ are ok_L and ok_C, respectively. □

3.3 Brouwerian Algebra

The relational data model is based on the powerset $\mathcal{P}(R)$ of a relation schema R. In fact, $\mathcal{P}(R)$ is a powerset algebra with partial order $\subseteq$, set union $\cup$, set intersection $\cap$ and set difference $-$. We will now extend these operations to nested attributes. The inclusion order $\subseteq$ has already been generalised by the subattribute relationship $\leq$. The finite set $Sub(N)$ of *subattributes* of N is $Sub(N) = \{M \mid M \leq N\}$. Lemma 4 shows that the restriction of $\leq$ to $Sub(N)$ defines a poset.

We study the algebraic structure of the poset $(Sub(N), \leq)$. A *Brouwerian algebra* [McKinsey and Tarski, 1946] is a lattice $(L, \sqsubseteq, \sqcup, \sqcap, \dot{-}, 1)$ with top element 1 and a binary operation $\dot{-}$ which satisfies $a \dot{-} b \sqsubseteq c$ iff $a \sqsubseteq b \sqcup c$ for all $c \in L$. In this case, the operation $\dot{-}$ is called the *pseudo-difference*. The *Brouwerian complement* $\neg a$ of $a \in L$ is then defined by $\neg a = 1 \dot{-} a$. A Brouwerian algebra is also called a co-Heyting algebra or a dual Heyting algebra. The system of all closed subsets of a topological space is a well-known Brouwerian algebra, see [McKinsey and Tarski, 1946]. The definition of the subattribute relationship $\leq$ completely determines the operations of join, meet and pseudo-difference. The following theorem generalises the fact that $(\mathcal{P}(R), \subseteq, \cup, \cap, -, \emptyset, R)$ is a Boolean algebra for a relation schema R in the relational data model. It is a simple consequence of the following observations: $Sub(\lambda_A)$ and $Sub(\lambda_L)$ are isomorphic to the Boolean algebra of order 0, $Sub(A)$ is isomorphic to the Boolean algebra of order 1, $Sub(L(N_1, \ldots, N_k))$ is isomorphic to the directed product of $Sub(N_1), \ldots, Sub(N_k)$, $Sub(L[N])$ is isomorphic to $Sub(N)$ augmented by a new minimum, and $Sub(L(N_1 \oplus \ldots \oplus N_k))$ is isomorphic to the directed product of $Sub(N_1), \ldots, Sub(N_k)$ augmented by a new minimum. Since Brouwerian algebras are closed under (finite) directed products and augmentations of a new bottom element [McKinsey and Tarski, 1946] this shows that $(Sub(N), \leq)$ carries the structure of a Brouwerian algebra. Notice that the use of different null

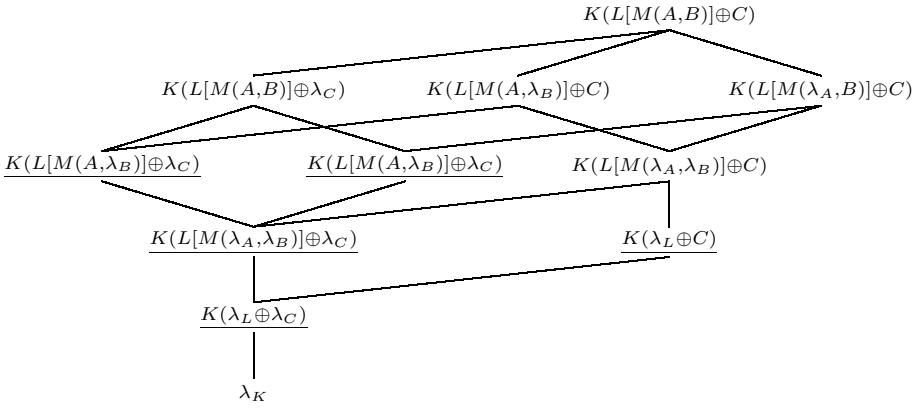

Figure 1: Brouwerian algebra of $K(L[M(A,B)]\oplus C)$

attributes resulted in a partial order of subattributes that is different from previous approaches. However, we still obtain a Brouwerian algebra, cf. [Hartmann et al., 2005; Hartmann et al., 2006a; Hartmann et al., 2006b].

Theorem 5. *$(Sub(N),\leq,\sqcup_N,\sqcap_N,\dot{-}_N,N)$ forms a Brouwerian algebra for every $N\in\mathcal{N}$.* □

The nested attribute N is the top element of $(Sub(N),\leq)$. The *bottom element* λ_N of $Sub(N)$ is given by

$$\lambda_N=\begin{cases}\lambda_L & \text{, if } N=\lambda_L\\ \lambda_A & \text{, if } N=A\in\mathcal{U}\\ L(\lambda_{N_1},\ldots,\lambda_{N_k}) & \text{, if } N=L(N_1,\ldots,N_k)\\ \lambda_L & \text{, if } N=L[M]\\ \lambda_L & \text{, if } N=L(N_1\oplus\cdots\oplus N_k)\end{cases}$$

The Brouwerian algebra for $K(L[M(A,B)]\oplus C)$ is illustrated in Figure 1.

If the context allows, we omit the index N from the operations $\sqcup_N,\sqcap_N,\dot{-}_N$ and from λ_N.

Recall that an element a of a lattice with bottom element 0 is called *join-irreducible* if and only if $a\neq 0$ and if $a=b\sqcup c$ holds for any elements b and c, then $a=b$ or $a=c$ [Graetzer, 1998]. The set of join-irreducible elements of $(Sub(N),\leq,\sqcup,\sqcap,\lambda_N)$ is denoted by $\mathcal{J}(N)$. We refer to elements of $\mathcal{J}(N)$ as join-irreducible elements of N. For instance, the join-irreducibles of $K(L[M(A,B)]\oplus C)$ are underlined in Figure 1.

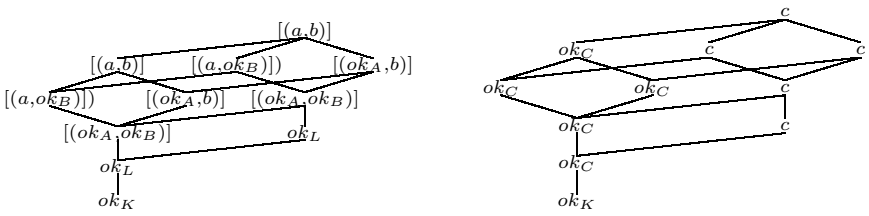

Figure 2: Projections of elements $[(a, b)]$ and c from $dom(K(L[M(A, B)] \oplus C))$

As an example for projections of actual domain elements consider Figure 2 in which the list $[(a, b)]$ and c from the domain of $K(L[M(A, B)] \oplus C)$ are projected on the subattributes of $K(L[M(A, B)] \oplus C)$.

3.4 The Definition of Weak Functional Dependencies and its Justification

In this part we will first introduce and illustrate weak functional dependencies on nested attributes. It is our aim to generalise Theorem 3 to the presence of record, list and disjoint union type. Therefore, we have to study the interactions between the structure of nested attributes and weak functional dependencies first.

Definition 6. A *weak functional dependency* (wFD) on the nested attribute N is an expression $\mathcal{X} \rightarrow_w \mathcal{Y}$ where $\mathcal{X}, \mathcal{Y} \subseteq \mathcal{J}(N)$. An instance $r \subseteq dom(N)$ *satisfies* the wFD $\mathcal{X} \rightarrow_w \mathcal{Y}$ on N, denoted by $\models_r \mathcal{X} \rightarrow_w \mathcal{Y}$, if and only if for every distinct pair of tuples in r that agree on each of the elements in $\mathcal{X}$, the tuples also agree on at least one element in $\mathcal{Y}$. That is, $\models_r \mathcal{X} \rightarrow_w \mathcal{Y}$ if and only if for all distinct $t_1, t_2 \in r$ such that for all $X \in \mathcal{X}$, $\pi^N_X(t_1) = \pi^N_X(t_2)$ holds, there is some $Y \in \mathcal{Y}$ such that $\pi^N_Y(t_1) = \pi^N_Y(t_2)$ holds. □

It is not obvious that Definition 6 takes the presence of our type constructors into account. For instance, in the presence of set- or multiset constructor it is insufficient to consider only join-irreducible subattributes. In that case, distinct tuples may agree on the projections to all join-irreducibles [Hartmann and Link, 2008; Hartmann et al., 2006a]. However, the next lemma shows that this situation cannot occur in the presence of record, list, and disjoint union constructor, i.e., distinct nested tuples disagree on at least some (maximal) join-irreducible.

Lemma 7. *Let N be a nested attribute, $X, Y \in Sub(N)$ and $t_1, t_2 \in dom(N)$. If $\pi^N_X(t_1) = \pi^N_X(t_2)$ and $\pi^N_Y(t_1) = \pi^N_Y(t_2)$, then $\pi^N_{X \sqcup Y}(t_1) = \pi^N_{X \sqcup Y}(t_2)$.*

Proof. We proceed by induction on the structure of N. The cases where $X \leq Y$, $Y \leq X$, and N is a record- or list-valued attribute are done exactly as in the proof of [Hartmann et al., 2006b, Lemma 3.1].

It remains to consider the case where $N = L(N_1 \oplus \cdots \oplus N_k)$ denotes a union-valued attribute. The hypothesis tells us that for all $i = 1, \ldots, k$, for all $X_i, Y_i \in Sub(N_i)$ and for all $t_1, t_2 \in dom(N_i)$ such that $\pi^{N_i}_{X_i}(t_1) = \pi^{N_i}_{X_i}(t_2)$ and $\pi^{N_i}_{Y_i}(t_1) = \pi^{N_i}_{Y_i}(t_2)$ holds, we also have $\pi^{N_i}_{X_i \sqcup Y_i}(t_1) = \pi^{N_i}_{X_i \sqcup Y_i}(t_2)$. It remains to consider the case where $X = L(X_1 \oplus \cdots \oplus X_k)$ and $Y = L(Y_1 \oplus \cdots \oplus Y_k)$. Let $t_1, t_2 \in dom(N)$ such that $\pi^N_X(t_1) = \pi^N_X(t_2)$ and $\pi^N_Y(t_1) = \pi^N_Y(t_2)$. It follows that for some $i \in \{1, \ldots, k\}$ we have $t_1, t_2 \in dom(N_i)$, $\pi^{N_i}_{X_i}(t_1) = \pi^{N_i}_{X_i}(t_2)$ and $\pi^{N_i}_{Y_i}(t_1) = \pi^{N_i}_{Y_i}(t_2)$. Consequently, we conclude by hypothesis that

$$\pi^N_{X \sqcup Y}(t_1) = \pi^{N_i}_{X_i \sqcup Y_i}(t_1) = \pi^{N_i}_{X_i \sqcup Y_i}(t_2) = \pi^N_{X \sqcup Y}(t_2).$$

This concludes the proof. □

In the literature there are different approaches toward defining dependencies in complex-value databases, e.g. in [Arenas and Libkin, 2004; Hara and Davidson, 1999; Vincent et al., 2004; Weddell, 1992]. We have compared our framework to such approaches in previous work [Hartmann and Link, 2006; Hartmann et al., 2006a].

3.5 Trivial Weak Functional Dependencies

Before we can attempt to generalise Theorem 3 we need to study the interaction of weak functional dependencies with the structure imposed by nested attributes. In fact, there are weak functional dependencies that are satisfied by every instance of some nested attributes. It is the goal of this section to describe such trivial weak functional dependencies by syntactical means.

We begin with an example that illustrates why weak functional dependencies trivially appear in the presence of unions.

Example 8. Let $N = Nat(Odd \oplus Even)$ where *dom(Odd)* are the odd positive integers and *dom(Even)* are the even non-negative integers. Then

$$\mathcal{J}(N) = \{Nat(Odd \oplus \lambda_{Even}), Nat(\lambda_{Odd} \oplus Even), Nat(\lambda_{Odd} \oplus \lambda_{Even})\}.$$

Take two distinct $n, m \in dom(N)$ such that

$$\pi^N_{Nat(\lambda_{Odd} \oplus \lambda_{Even})}(n) = \pi^N_{Nat(\lambda_{Odd} \oplus \lambda_{Even})}(m).$$

That is, either both n and m are even or both n and m are odd. In the first case we have

$$\pi^N_{Nat(Odd \oplus \lambda_{Even})}(n) = \pi^N_{Nat(Odd \oplus \lambda_{Even})}(m),$$

and in the second case we have

$$\pi^N_{Nat(\lambda_{Odd} \oplus Even)}(n) = \pi^N_{Nat(\lambda_{Odd} \oplus Even)}(m).$$

That is, the wFD

$$Nat(\lambda_{Odd} \oplus \lambda_{Even}) \rightarrow_w \{Nat(\lambda_{Odd} \oplus Even), Nat(Odd \oplus \lambda_{Even})\}$$

is satisfied by every instance $r \subseteq dom(N)$, i.e., the wFD is in fact trivial. □

Notice the structure of the nested attributes that occur in the trivial wFD

$$Nat(\lambda_{Odd} \oplus \lambda_{Even}) \rightarrow_w \{Nat(\lambda_{Odd} \oplus Even), Nat(Odd \oplus \lambda_{Even})\}$$

in Example 8. We will need to characterise those nested attributes syntactically before we can attempt to find a sound and complete set of inference rules.

The following definition will enable us to syntactically capture all left-hand sides of those trivial weak functional dependencies that are similar to the ones we have just encountered in Example 8.

Definition 8. Let N be a nested attribute. The set $N^{\oplus}$ of *$\oplus$-join-irreducibles of N* is recursively defined by

- if $N = \lambda_A$ for some $A \in \mathcal{U}$ or $N = \lambda_L$ for some $L \in \mathcal{L}$, then $N^{\oplus} = \emptyset$,
- if $N = A \in \mathcal{U}$, then $N^{\oplus} = \emptyset$,
- if $N = L[M]$, then $N^{\oplus} = \emptyset$,
- if $N = L(N_1, \ldots, N_k)$, then

$$N^{\oplus} = \bigcup_{i=1}^{k} \{L(\lambda_{N_1}, \ldots, M_i, \ldots, \lambda_{N_k}) \mid M_i \in N_i^{\oplus}\},$$

- if $N = L(N_1 \oplus \cdots \oplus N_k)$, then

$$N^{\oplus} = \{L(\lambda_{N_1} \oplus \cdots \oplus \lambda_{N_k})\} \cup \bigcup_{i=1}^{k} \{L(\lambda_{N_1} \oplus \cdots \oplus M_i \oplus \cdots \oplus \lambda_{N_k}) \mid M_i \in N_i^{\oplus}\}.$$

□

Intuitively, Definition 8 identifies those union-valued nested attributes that are generated from null attributes only (see item five) as $\oplus$-join-irreducibles, but also those complex union-valued nested attribute (item five again) and record-valued nested attribute (item four) that are generated from null attributes and $\oplus$-join-irreducibles.

Example 9. Let $N = L(K(M(A \oplus B \oplus C), D) \oplus O[P(E \oplus F)])$. Then $N^{\oplus}$ consists of $L(K(\lambda_M, \lambda_D) \oplus \lambda_O)$ and $L(K(M(\lambda_A \oplus \lambda_B \oplus \lambda_C), \lambda_D) \oplus \lambda_O)$. □

List-valued attributes do not define any $\oplus$-*join-irreducibles*, cf. Definition 8 (item three). The following examples illustrate the reason. That is, list-valued attributes do not exhibit those trivial weak functional dependencies we have encountered before.

Example 10. Let $N = Seq[Nat(Odd \oplus Even)]$. The wFD

$$Seq[Nat(\lambda_{Odd} \oplus \lambda_{Even})] \rightarrow_w \{Seq[Nat(Odd \oplus \lambda_{Even})], Seq[Nat(\lambda_{Odd} \oplus Even)]\}$$

is not trivial. Consider for instance the two distinct lists $t_1 = [2, 3]$ and $t_2 = [4, 5]$. We see that

$$\pi^N_{Seq[Nat(\lambda_{Odd} \oplus \lambda_{Even})]}(t_1) = [ok_{Even}, ok_{Odd}] = \pi^N_{Seq[Nat(\lambda_{Odd} \oplus \lambda_{Even})]}(t_2),$$

but

$$\pi^N_{Seq[Nat(Odd \oplus \lambda_{Even})]}(t_1) = [ok_{Even}, 3] \neq [ok_{Even}, 5] = \pi^N_{Seq[Nat(Odd \oplus \lambda_{Even})]}(t_2)$$

and

$$\pi^N_{Seq[Nat(\lambda_{Odd} \oplus Even)]}(t_1) = [2, ok_{Odd}] \neq [4, ok_{Odd}] = \pi^N_{Seq[Nat(\lambda_{Odd} \oplus Even)]}(t_2).$$

The instance $r = \{t_1, t_2\}$ shows therefore that the wFD above is not trivial. □

Example 11. Consider the following list-valued attribute

$$\text{Nucleotide[Base(Purine} \oplus \text{Pyrimidine)]}$$

where

$$dom(\text{Purine}) = \{\text{A(denine), G(uanine)}\} \text{ and}$$
$$dom(\text{Pyrimidine}) = \{\text{C(ytosine),T(hymine)}\}.$$

This database schema describes nucleotide sequences. The database consisting of the four sequences

[ATG TCG GCG GGA]
[ACG TCG GCG GGA]
[TAC AGT CGT]
[TAC GAT CAT]

satisfies the weak functional dependency

$$\text{Nucleotide}[\lambda_{\text{Base}})] \rightarrow_w$$
$$\{\text{Nucleotide[Base(Purine} \oplus \lambda_{\text{Pyrimidine}})], \text{Nucleotide[Base}(\lambda_{\text{Purine}} \oplus \text{Pyrimidine)]}\} .$$

Notice that only the first two tuples and the last two tuples match on

$$\text{Nucleotide}[\lambda_{\text{Base}})],$$

but no other pairs of distinct tuples. There is no case of different Purine occurring in the same position of the first two tuples, i.e., the first two tuples also match on

$$\text{Nucleotide}[\text{Base}(\text{Purine} \oplus \lambda_{\text{Pyrimidine}}).]$$

There is also no case of different Pyrimidine occurring in the same position of the last two tuples, i.e., the last two tuples also match on

$$\text{Nucleotide}[\text{Base}(\lambda_{\text{Purine}} \oplus \text{Pyrimidine})].$$

However, the database consisting of the two sequences

[TAC AGT CGT]
[CAT GAT CGT]

does not satisfy this weak functional dependency. In fact, the two sequences match on

$$\text{Nucleotide}[\lambda_{\text{Base}})]$$

but do not match on neither

$$\text{Nucleotide}[\text{Base}(\text{Purine} \oplus \lambda_{\text{Pyrimidine}})] \text{ nor}$$
$$\text{Nucleotide}[\text{Base}(\lambda_{\text{Purine}} \oplus \text{Pyrimidine})].$$

□

In Definition 8 we have captured the nested attributes that occur on the left-hand side of certain trivial wFDs. It remains to identify the right-hand sides. Therefore, we look at an example first.

Example 12. Let $N = L(A \oplus K(M(B \oplus C), D))$. Then $N^{\oplus}$ consists of $O_1 = L(\lambda_A \oplus K(\lambda_M, \lambda_D))$ and $O_2 = L(\lambda_A \oplus K(M(\lambda_B \oplus \lambda_C), \lambda_D))$. The maximal join-irreducibles are $W = L(A \oplus K(\lambda_M, \lambda_D))$, $X = L(\lambda_A \oplus K(M(B \oplus \lambda_C), \lambda_D))$, $Y = L(\lambda_A \oplus K(M(\lambda_B \oplus C), \lambda_D))$ and $Z = L(\lambda_A \oplus K(\lambda_M, D))$. The situation is illustrated in Figure 3.

Let $t_1 = (b, d)$ and $t_2 = (b', d)$ be elements from $dom(N)$ where b, b' are distinct elements from $dom(B)$. It follows that $\pi^N_{O_2}(t_1) = (ok_B, ok_D) = \pi^N_{O_2}(t_2)$. We can also see that $\pi^N_Y(t_1) = (ok_B, ok_D) = \pi^N_Y(t_2)$. This is not a random case. We will see soon that such O_2, X, Y carry the trivial weak functional dependency $O_2 \rightarrow_w \{X, Y\}$.

However, the situation is subtle. Let $t_1 = (b, d)$ and $t_2 = (c, d)$ be elements from $dom(N)$ where $b \in dom(B)$ and $c \in dom(C)$. It follows that

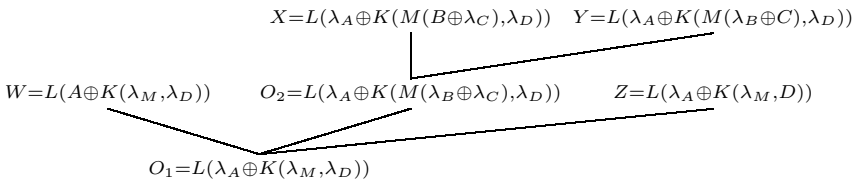

Figure 3: Branched Pairs of Join-Irreducibles

$\pi^N_{O_1}(t_1) = (ok_M, ok_D) = \pi^N_{O_1}(t_2)$. However, we can also see that $\pi^N_X(t_1) = (b, ok_D) \neq (ok_C, ok_D) = \pi^N_X(t_2)$ and $\pi^N_Y(t_1) = (ok_B, ok_D) \neq (c, ok_D) = \pi^N_Y(t_2)$. Hence, the weak functional dependency $O_1 \rightarrow_w \{X, Y\}$ is not trivial. □

The following definition enables us to identify the correct pairs X, Y with respect to an $\oplus$-join-irreducible O of N such that $O \rightarrow_w \{X, Y\}$ holds indeed in every nested relation over the underlying nested attribute.

Definition 9. Let $N \in \mathcal{N}$ be a nested attribute and $O \in N^{\oplus}$. A pair $X, Y \in \mathcal{J}(N)$ is called *branched with respect to* O if and only if $O < X, Y$ and at least one of the following conditions is satisfied

- $N = L(N_1 \oplus \cdots \oplus N_k)$, $O = L(\lambda_{N_1} \oplus \cdots \oplus O_i \oplus \cdots \oplus \lambda_{N_k})$ with $O_i \in N_i^{\oplus}$, $X = L(\lambda_{N_1} \oplus \cdots \oplus X_i \oplus \cdots \oplus \lambda_{N_k})$ and $Y = L(\lambda_{N_1} \oplus \cdots \oplus Y_i \oplus \cdots \oplus \lambda_{N_k})$ and $X_i, Y_i \in Sub(N_i)$ are branched with respect to O_i,
- $N = L(N_1, \ldots, N_k)$, $O = L(\lambda_{N_1}, \ldots, O_i, \ldots, \lambda_{N_k})$ with $O_i \in N_i^{\oplus}$, $X = L(\lambda_{N_1}, \ldots, X_i, \ldots, \lambda_{N_k}), Y = L(\lambda_{N_1}, \ldots, Y_i, \ldots, \lambda_{N_k})$ and $X_i, Y_i \in Sub(N_i)$ are branched with respect to O_i,
- $N = L(N_1 \oplus \cdots \oplus N_k)$, $O = L(\lambda_{N_1} \oplus \cdots \oplus \lambda_{N_k})$, $X \leq L(\lambda_{N_1} \oplus \cdots \oplus N_i \oplus \cdots \oplus \lambda_{N_k})$ and $Y \leq L(\lambda_{N_1} \oplus \cdots \oplus N_j \oplus \cdots \oplus \lambda_{N_k})$ with $i \neq j$. □

Intuitively, the pair $X, Y \in \mathcal{J}(N)$ is branched with respect to O if X and Y occur in different branches of the underlying union-valued attribute N (item three), or they are union-valued attributes generated from null attributes and nested attributes that are branched with respect to the same branch of O (item one) or they are record-valued attributes generated from null attributes and nested attributes that are branched with respect to the same branch of O (item two). List-valued attributes are not included in this definition as this kind of trivial weak functional dependency does not occur in the presence of lists, see Examples 10 and 11.

Example 13. Consider Example 12 and Figure 3 again. The pairs W, X, and W, Y, and W, Z and Y, Z and X, Z are O_1-branched. The pair X, Y is O_2-branched, but X, Y is not O_1-branched. □

We will now show that our definitions capture trivial weak functional dependencies. Indeed, the next lemma shows that branched pairs of join-irreducibles exhibit trivial weak functional dependencies. In fact, if two tuples agree on some $O \in N^{\oplus}$, then they also agree on one of the elements in any pair X, Y that is branched with respect to O.

Lemma 10. *Let $N \in \mathcal{N}$, $O \in N^{\oplus}$ and $X, Y \in \mathcal{J}(N)$ branched with respect to O. Let $t_1, t_2 \in dom(N)$ such that $\pi^N_O(t_1) = \pi^N_O(t_2)$. Then $\pi^N_X(t_1) = \pi^N_X(t_2)$ or $\pi^N_Y(t_1) = \pi^N_Y(t_2)$.*

Proof. We proceed by induction on the structure N of nested attributes. In the case where $N = \lambda_A$, $N = \lambda_L$, $N = A$ and $N = L[M]$ there is nothing to show since $N^{\oplus} = \emptyset$.

Let $N = L(N_1, \ldots, N_k)$ and $O = L(\lambda_{N_1}, \ldots, O_i, \ldots, \lambda_{N_k})$ where $O_i \in N_i^{\oplus}$. Let $X = L(\lambda_{N_1}, \ldots, X_i, \ldots, \lambda_{N_k})$ and $Y = L(\lambda_{N_1}, \ldots, Y_i, \ldots, \lambda_{N_k})$ such that $X_i, Y_i \in Sub(N_i)$ are branched with respect to O_i. Let $t_1 = (t_1^1, \ldots, t_1^k), t_2 = (t_2^1, \ldots, t_2^k) \in dom(N)$ such that $\pi^N_O(t_1) = \pi^N_O(t_2)$ holds. This implies, in particular, that $\pi^{N_i}_{O_i}(t_1^i) = \pi^{N_i}_{O_i}(t_2^i)$. However, we can conclude by hypothesis that $\pi^{N_i}_{X_i}(t_1^i) = \pi^{N_i}_{X_i}(t_2^i)$ or $\pi^{N_i}_{Y_i}(t_1^i) = \pi^{N_i}_{Y_i}(t_2^i)$ holds. This is just the same as saying that $\pi^N_X(t_1) = \pi^N_X(t_2)$ or $\pi^N_Y(t_1) = \pi^N_Y(t_2)$ holds.

Let $N = L(N_1 \oplus \cdots \oplus N_k), O = L(\lambda_{N_1} \oplus \cdots \oplus O_i \oplus \cdots \oplus \lambda_{N_k})$ where $O_i \in N_i^{\oplus}$. Let $X = L(\lambda_{N_1} \oplus \cdots \oplus X_i \oplus \cdots \oplus \lambda_{N_k})$ and $Y = L(\lambda_{N_1} \oplus \cdots \oplus Y_i \oplus \cdots \oplus \lambda_{N_k})$ and $X_i, Y_i \in Sub(N_i)$ are branched with respect to O_i. Let $t_1, t_2 \in dom(N)$ such that $\pi^N_O(t_1) = \pi^N_O(t_2)$. We distinguish between two cases. In the first case we have $t_1, t_2 \in dom(N_j)$ where $j \neq i$. In this case $\pi^N_X(t_1) = \pi^N_O(t_1) = \pi^N_O(t_2) = \pi^N_X(t_2)$ and $\pi^N_Y(t_1) = \pi^N_O(t_1) = \pi^N_O(t_2) = \pi^N_Y(t_2)$. In the remaining case we have $t_1, t_2 \in dom(N_i)$. This means that $\pi^{N_i}_{O_i}(t_1) = \pi^{N_i}_{O_i}(t_2)$ and the hypothesis shows that $\pi^{N_i}_{X_i}(t_1) = \pi^{N_i}_{X_i}(t_2)$ or $\pi^{N_i}_{Y_i}(t_1) = \pi^{N_i}_{Y_i}(t_2)$ holds. This, however, is just the same as $\pi^N_X(t_1) = \pi^N_X(t_2)$ or $\pi^N_Y(t_1) = \pi^N_Y(t_2)$, respectively. We do not have to consider other cases since we assume that $\pi^N_O(t_1) = \pi^N_O(t_2)$ holds.

Let $N = L(N_1 \oplus \cdots \oplus N_k), O = L(\lambda_{N_1} \oplus \cdots \oplus \lambda_{N_k}), X \leq L(\lambda_{N_1} \oplus \cdots \oplus N_i \oplus \cdots \oplus \lambda_{N_k})$ and $Y \leq L(\lambda_{N_1} \oplus \cdots \oplus N_j \oplus \cdots \oplus \lambda_{N_k})$ such that $i \neq j$. Let $t_1, t_2 \in dom(N)$ such that $\pi^N_O(t_1) = \pi^N_O(t_2)$ holds. This means that $t_1, t_2 \in dom(N_l)$ for some l with $1 \leq l \leq k$. If $l = i$, then $\pi^N_Y(t_1) = \pi^N_Y(t_2)$. If $l = j$, then $\pi^N_X(t_1) = \pi^N_X(t_2)$. Finally, if $l \neq i$ and $l \neq j$, then $\pi^N_X(t_1) = \pi^N_X(t_2)$ and $\pi^N_Y(t_1) = \pi^N_Y(t_2)$. This completes the proof. □

Example 14. Consider Example 12 again. The first choice of tuples t_1, t_2 reflects Lemma 10. The second choice of tuples t_1, t_2 illustrates that the notion of branching captures the right pairs of join-irreducibles. □

$$\frac{}{\mathcal{X} \to_w \mathcal{Y}}\ {\scriptstyle \exists X \in \mathcal{X}.\exists Y \in \mathcal{Y}.Y \leq X} \quad \text{(Axiom 1)}$$

$$\frac{}{\mathcal{J}(N) \to_w \emptyset} \quad \text{(Axiom 2)}$$

$$\frac{\mathcal{X} \to_w \mathcal{Y}}{\mathcal{X} \to_w \mathcal{Z}}\ {\scriptstyle \mathcal{Y} \subseteq \mathcal{Z}} \quad \text{(Rule 1)}$$

$$\frac{\mathcal{X} \to_w \mathcal{Y}}{\mathcal{Z} \to_w \mathcal{Y}}\ {\scriptstyle \mathcal{X} \subseteq \mathcal{Z}} \quad \text{(Rule 2)}$$

$$\frac{\mathcal{X}^{\downarrow} \to_w \mathcal{Y}}{\mathcal{X} \to_w \mathcal{Y}} \quad \text{(Rule 3)}$$

$$\frac{\mathcal{X} \to_w \mathcal{Y}^{\uparrow}}{\mathcal{X} \to_w \mathcal{Y}} \quad \text{(Rule 4)}$$

$$\frac{\{\mathcal{X}\mathcal{V}^{\downarrow} \to_w \mathcal{Y}(\mathcal{U} - \mathcal{V}^{\downarrow})\}\ {\scriptstyle \mathcal{V} \subseteq \mathcal{U}}}{\mathcal{X} \to_w \mathcal{Y}} \quad \text{(Rule 5)}$$

$$\frac{}{O \to_w \{X, Y\}}\ {\scriptstyle O \in N^{\oplus}, X, Y \text{ are } O\text{-branched}} \quad \text{(Axiom 3)}$$

Table 3: An Axiomatisation of wFDs in complex-value databases

3.6 An Axiomatisation

We will use this section to establish an axiomatisation for the implication of weak functional dependencies on nested attributes generated by record, list and disjoint union constructor. We outline the completeness proof in this section, but devote the next section to complete our description of generating the two tuple database instance required for the completeness argument.

Before we generalise Theorem 3 from relational databases to complex-value databases we introduce some more notation. Let $\mathcal{X} \subseteq \mathcal{J}(N)$. The *downward closure* of $\mathcal{X}$ with respect to $\leq$ is denoted by $\mathcal{X}^{\downarrow} = \{Y \in \mathcal{J}(N) \mid Y \leq X \text{ for some } X \in \mathcal{X}\}$. Correspondingly, $\mathcal{X}^{\uparrow} = \{Y \in \mathcal{J}(N) \mid X \leq Y \text{ for some } X \in \mathcal{X}\}$ denotes the *upward closure* of $\mathcal{X}$ with respect to $\leq$. In particular, if $\mathcal{X}^{\downarrow}$ and $\mathcal{X}^{\uparrow}$ are non-empty, then they denote *ideals* and *filters*, respectively.

Theorem 11. *Let $\mathfrak{S}$ denote the set of inference rules from Table 3. Then $\mathfrak{S}$ is sound and complete for the implication of wFDs in complex-value databases.*

Proof. Our first step is to verify the soundness of the inference rules. Consider *Axiom 1.* Let N be some nested attribute and $r \subseteq dom(N)$ be arbitrary. Assume there are any distinct $t_1, t_2 \in r$ such that $\pi^N_X(t_1) = \pi^N_X(t_2)$ holds for all $X \in \mathcal{X}$. For some $X \in \mathcal{X}$ there is some $Y \in \mathcal{Y}$ such that $Y \leq X$ holds. Since $X \in \mathcal{X}$ we have $\pi^N_X(t_1) = \pi^N_X(t_2)$, but since $Y \leq X$ we also have $\pi^N_Y(t_1) = \pi^N_Y(t_2)$ by definition of the projection function. This means that r satisfies $\mathcal{X} \to_w \mathcal{Y}$.

Consider *Axiom 2.* Let N be some nested attribute and $r \subseteq dom(N)$ be arbitrary. Any distinct $t_1, t_2 \in r$ must satisfy $\pi^N_X(t_1) \neq \pi^N_X(t_2)$ for some $X \in \mathcal{J}(N)$ by contraposition of Lemma 7. This means that r satisfies $\mathcal{J}(N) \to_w \emptyset$.

Consider *Axiom 3*. Let N be some nested attribute and $r \subseteq dom(N)$ be arbitrary, and $O \in N^{\oplus}$. Assume there are any distinct $t_1, t_2 \in r$ such that $\pi^N_O(t_1) = \pi^N_O(t_2)$ holds. Let X, Y be any pair of join-irreducibles that are branched with respect to O. Lemma 10 shows that $\pi^N_X(t_1) = \pi^N_X(t_2)$ or $\pi^N_Y(t_1) = \pi^N_Y(t_2)$ holds. This means that r satisfies $O \rightarrow_w \{X, Y\}$.

Consider *Rule 1*. Let N be some nested attribute and $r \subseteq dom(N)$ be arbitrary such that r satisfies $\mathcal{X} \rightarrow_w \mathcal{Y}$. Assume there are any distinct $t_1, t_2 \in r$ such that $\pi^N_X(t_1) = \pi^N_X(t_2)$ holds for all $X \in \mathcal{X}$. It follows that $\pi^N_Y(t_1) = \pi^N_Y(t_2)$ holds for some $Y \in \mathcal{Y}$ since r satisfies $\mathcal{X} \rightarrow_w \mathcal{Y}$ by assumption. However, since $\mathcal{Y} \subseteq \mathcal{Z}$ holds by assumption it follows that $Y \in \mathcal{Z}$, too. Consequently, $\pi^N_Y(t_1) = \pi^N_Y(t_2)$ holds for some $Y \in \mathcal{Z}$. We conclude that r satisfies $\mathcal{X} \rightarrow_w \mathcal{Z}$.

Consider *Rule 2*. Let N be some nested attribute and $r \subseteq dom(N)$ be arbitrary such that r satisfies $\mathcal{X} \rightarrow_w \mathcal{Y}$. Assume there are any distinct $t_1, t_2 \in r$ such that $\pi^N_Z(t_1) = \pi^N_Z(t_2)$ holds for all $Z \in \mathcal{Z}$. Since $\mathcal{X} \subseteq \mathcal{Z}$ holds by assumption it follows that $\pi^N_X(t_1) = \pi^N_X(t_2)$ holds for all $X \in \mathcal{X}$. However, r satisfies $\mathcal{X} \rightarrow_w \mathcal{Y}$, i.e.,$\pi^N_Y(t_1) = \pi^N_Y(t_2)$ holds for some $Y \in \mathcal{Y}$. We conclude that r satisfies $\mathcal{Z} \rightarrow_w \mathcal{Y}$.

Consider *Rule 3*. Let N be some nested attribute and $r \subseteq dom(N)$ be arbitrary such that r satisfies $\mathcal{X}^{\downarrow} \rightarrow_w \mathcal{Y}$. Assume there are any distinct $t_1, t_2 \in r$ such that $\pi^N_X(t_1) = \pi^N_X(t_2)$ holds for all $X \in \mathcal{X}$. It follows that $\pi^N_Z(t_1) = \pi^N_Z(t_2)$ holds for all $Z \in \mathcal{X}^{\downarrow}$ since for every $Z \in \mathcal{X}^{\downarrow}$ there is some $X \in \mathcal{X}$ such that $Z \leq X$ holds. Since r satisfies $\mathcal{X}^{\downarrow} \rightarrow_w \mathcal{Y}$ we conclude that $\pi^N_Y(t_1) = \pi^N_Y(t_2)$ holds for some $Y \in \mathcal{Y}$. Hence, r satisfies $\mathcal{X} \rightarrow_w \mathcal{Y}$.

Consider *Rule 4*. Let N be some nested attribute and $r \subseteq dom(N)$ be arbitrary such that r satisfies $\mathcal{X} \rightarrow_w \mathcal{Y}^{\uparrow}$. Assume there are any distinct $t_1, t_2 \in r$ such that $\pi^N_X(t_1) = \pi^N_X(t_2)$ holds for all $X \in \mathcal{X}$. It follows that there is some $Z \in \mathcal{Y}^{\uparrow}$ such that $\pi^N_Z(t_1) = \pi^N_Z(t_2)$ holds since r satisfies $\mathcal{X} \rightarrow_w \mathcal{Y}^{\uparrow}$. However, that means there is some $Y \in \mathcal{Y}$ with $Y \leq Z$ by definition of $\mathcal{Y}^{\uparrow}$. We conclude that $\pi^N_Y(t_1) = \pi^N_Y(t_2)$ holds for some $Y \in \mathcal{Y}$ by definition of the projection function. Hence, r satisfies $\mathcal{X} \rightarrow_w \mathcal{Y}$.

Consider *Rule 5*. Let N be some nested attribute and $r \subseteq dom(N)$ be some arbitrary instance that satisfies all wFDs $\mathcal{X}\mathcal{V}^{\downarrow} \rightarrow_w \mathcal{Y}(\mathcal{U} - \mathcal{V}^{\downarrow})$ for all $\mathcal{V} \subseteq \mathcal{U}$ of some $\mathcal{U} \subseteq \mathcal{J}(N)$. Assume there are some distinct $t_1, t_2 \in r$ such that $\pi^N_A(t_1) = \pi^N_A(t_2)$ for all $A \in \mathcal{X}$. Since r satisfies the wFD $\mathcal{X} \rightarrow_w \mathcal{Y}\mathcal{U}$ there is some $Y_1 \in \mathcal{Y}\mathcal{U}$ with $\pi^N_{Y_1}(t_1) = \pi^N_{Y_1}(t_2)$. If $Y_1 \in \mathcal{Y}$, then there is nothing more to show. Otherwise, $Y_1 \in \mathcal{U}$. Since r satisfies the wFD $\mathcal{X}Y_1^{\downarrow} \rightarrow_w \mathcal{Y}(\mathcal{U} - Y_1^{\downarrow})$ there is some $Y_2 \in \mathcal{Y}(\mathcal{U} - Y_1^{\downarrow})$ with $\pi^N_{Y_2}(t_1) = \pi^N_{Y_2}(t_2)$. If $Y_2 \in \mathcal{Y}$, then there is nothing more to show. Otherwise, we continue this line of reasoning until we have found that $\pi^N_X(t_1) = \pi^N_X(t_2)$ holds for all $X \in \mathcal{X}\mathcal{U}$. Since r also satisfies $\mathcal{X}\mathcal{U} \rightarrow_w \mathcal{Y}$ there is some $Y \in \mathcal{Y}$ with $\pi^N_Y(t_1) = \pi^N_Y(t_2)$. Consequently, r satisfies also the wFD $\mathcal{X} \rightarrow_w \mathcal{Y}$.

The soundness of the inference system follows from the soundness of its in-

dividual rules by a simple induction on the length of the inference.

We will now verify the completeness of the inference rules for the implication of weak functional dependencies in complex-value databases. We need to show that $\Sigma^* \subseteq \Sigma^+$ holds. Let $\mathcal{X} \rightarrow_w \mathcal{Y} \notin \Sigma^+$. We will show that $\mathcal{X} \rightarrow_w \mathcal{Y} \notin \Sigma^*$ holds as well by constructing a two-element instance $r \subseteq dom(N)$ that satisfies all wFDs in Σ but violates $\mathcal{X} \rightarrow_w \mathcal{Y}$.

Since $\mathcal{X} \rightarrow_w \mathcal{Y} \notin \Sigma^+$ we conclude that $\mathcal{X}^\downarrow \rightarrow_w \mathcal{Y} \notin \Sigma^+$ as otherwise $\mathcal{X} \rightarrow_w \mathcal{Y} \in \Sigma^+$ holds by *Rule 3*. Moreover, we conclude that $\mathcal{X}^\downarrow \rightarrow_w \mathcal{Y}^\uparrow \notin \Sigma^+$ as otherwise $\mathcal{X}^\downarrow \rightarrow_w \mathcal{Y} \in \Sigma^+$ by *Rule 4*. According to *Axiom 1* the following holds since $\mathcal{X}^\downarrow \rightarrow_w \mathcal{Y}^\uparrow \notin \Sigma^+$: $\forall Y \in \mathcal{Y}^\uparrow . \forall X \in \mathcal{X}^\downarrow . Y \not\leq X$. That is, no $Y \in \mathcal{Y}^\uparrow$ is a subattribute of any $X \in \mathcal{X}^\downarrow$. Moreover, $\mathcal{J}(N) - \mathcal{X}^\downarrow \neq \emptyset$ as otherwise $\mathcal{Y}^\uparrow = \emptyset$ held, and therefore also $\mathcal{X}^\downarrow \rightarrow_w \mathcal{Y}^\uparrow \in \Sigma^+$ by *Axiom 2*.

We define $\mathcal{U} := \mathcal{J}(N) - (\mathcal{X}^\downarrow \cup \mathcal{Y}^\uparrow)$. Then there is some $\mathcal{V} \subseteq \mathcal{U}$ such that $\mathcal{X}^\downarrow \mathcal{V}^\downarrow \rightarrow_w \mathcal{Y}^\uparrow(\mathcal{U} - \mathcal{V}^\downarrow) \notin \Sigma^+$ as otherwise $\mathcal{X}^\downarrow \rightarrow_w \mathcal{Y}^\uparrow \in \Sigma^+$ by *Rule 5*.

Assume that $\mathcal{Y}^\uparrow(\mathcal{U} - \mathcal{V}^\downarrow) = \emptyset$, i.e., $\mathcal{Y}^\uparrow = \emptyset$ and $\mathcal{U} \subseteq \mathcal{V}^\uparrow$. By definition of $\mathcal{U}$ we then have $\mathcal{J}(N) - \mathcal{X}^\downarrow = \mathcal{U} \subseteq \mathcal{V}^\downarrow$ and, therefore, $\mathcal{J}(N) \subseteq \mathcal{X}^\downarrow \mathcal{V}^\downarrow \subseteq \mathcal{J}(N)$. However, this would imply that $\mathcal{X}^\downarrow \mathcal{V}^\downarrow \rightarrow_w \mathcal{Y}^\uparrow(\mathcal{U} - \mathcal{V}^\downarrow) \in \Sigma^+$, a contradiction. Hence, $\mathcal{Y}^\uparrow(\mathcal{U} - \mathcal{V}^\downarrow) \neq \emptyset$.

Assume there is some $O \in N^\oplus$ such that $O \in \mathcal{X}^\downarrow \mathcal{V}^\downarrow$ and there are $X, Y \in \mathcal{J}(N)$ that are branched with respect to O and $X, Y \in \mathcal{J}(N) - (\mathcal{X}^\downarrow \mathcal{V}^\downarrow) = \mathcal{Y}^\uparrow(\mathcal{U} - \mathcal{V}^\downarrow)$. In this case we conclude that $O \rightarrow_w \{X, Y\} \in \Sigma^+$ by *Axiom 3*. According to our assumptions we conclude that $\mathcal{X}^\downarrow \mathcal{V}^\downarrow \rightarrow_w \{X, Y\} \in \Sigma^+$ by *Rule 2* and $\mathcal{X}^\downarrow \mathcal{V}^\downarrow \rightarrow_w \mathcal{Y}^\uparrow(\mathcal{U} - \mathcal{V}^\downarrow) \in \Sigma^+$ by *Rule 1*, a contradiction. Consequently, the $\leq$-downward closure $\mathcal{X}^\downarrow \mathcal{V}^\downarrow$ has the following property: for all $O \in N^\oplus$ with $O \in \mathcal{X}^\downarrow \mathcal{V}^\downarrow$ and for all pairs $X, Y \in \mathcal{J}(N)$ that are branched with respect to O it follows that $X \in \mathcal{X}^\downarrow \mathcal{V}^\downarrow$ or $Y \in \mathcal{X}^\downarrow \mathcal{V}^\downarrow$.

According to Lemma 12 we can define an instance $r = \{t_1, t_2\} \subseteq dom(N)$ such that for all $X \in \mathcal{J}(N)$ we have

$$\pi_X^N(t_1) = \pi_X^N(t_2) \quad \text{if and only if} \quad X \in \mathcal{X}^\downarrow \mathcal{V}^\downarrow \text{ holds.}$$

Notice that the tuples t_1 and t_2 are distinct since $\mathcal{J}(N) - (\mathcal{X}^\downarrow \mathcal{V}^\downarrow) = \mathcal{Y}^\uparrow(\mathcal{U} - \mathcal{V}^\downarrow) \neq \emptyset$. We observe that r does not satisfy $\mathcal{X} \rightarrow_w \mathcal{Y}$ since $\mathcal{X} \subseteq \mathcal{X}^\downarrow \mathcal{V}^\downarrow$ and $\mathcal{Y} \subseteq \mathcal{Y}^\uparrow(\mathcal{U} - \mathcal{V}^\downarrow)$. It remains to show that r satisfies all weak functional dependencies in Σ.

Let $\mathcal{W} \rightarrow_w \mathcal{Z} \in \Sigma$. Suppose $\pi_W^N(t_1) = \pi_W^N(t_2)$ holds for all $W \in \mathcal{W}$. Otherwise it is immediate that r satisfies $\mathcal{W} \rightarrow_w \mathcal{Z}$. According to the definition of r it follows that $\mathcal{W} \subseteq \mathcal{X}^\downarrow \mathcal{V}^\downarrow$. This, however, implies that $\mathcal{X}^\downarrow \mathcal{V}^\downarrow \rightarrow_w \mathcal{Z} \in \Sigma^+$ by *Rule 2*. Assume that $\mathcal{Z} \subseteq \mathcal{Y}^\uparrow(\mathcal{U} - \mathcal{V}^\downarrow)$. It follows that $\mathcal{X}^\downarrow \mathcal{V}^\downarrow \rightarrow_w \mathcal{Y}^\uparrow(\mathcal{U} - \mathcal{V}^\downarrow) \in \Sigma^+$ by *Rule 3*, a contradiction to our assumptions. Consequently, $\mathcal{Z} \not\subseteq \mathcal{Y}^\uparrow(\mathcal{U} - \mathcal{V}^\downarrow) = \mathcal{J}(N) - (\mathcal{X}^\downarrow \mathcal{V}^\downarrow)$. We conclude that $\mathcal{Z} \cap \mathcal{X}^\downarrow \mathcal{V}^\downarrow \neq \emptyset$ and, therefore, $\pi_Z^N(t_1) = \pi_Z^N(t_2)$ holds for some $Z \in \mathcal{Z}$ according to the definition of r. This means that r satisfies $\mathcal{W} \rightarrow_w \mathcal{Z}$. □

In order to complete the proof of Theorem 11 we must show that it is possible to generate the two-element relation $r = \{t_1, t_2\} \subseteq dom(N)$ whose two nested tuples agree on precisely those subattributes that belong to a $\leq$-downward closure.

Example 15. Consider Example 8 again. The attributes $X = Nat(Odd \oplus \lambda_{Even})$ and $Y = Nat(\lambda_{Odd} \oplus Even)$ are branched with respect to $O = Nat(\lambda_{Odd} \oplus \lambda_{Even})$. Let $\mathcal{X} = \{O\}$ be a $\leq$-ideal in $\mathcal{J}(N)$. Then Lemma 10 shows that there are no tuples $t_1, t_2 \in dom(N)$ such that $\pi^N_O(t_1) = \pi^N_O(t_2)$, but $\pi^N_X(t_1) \neq \pi^N_X(t_2)$ and $\pi^N_Y(t_1) \neq \pi^N_Y(t_2)$. □

Example 15 shows that the construction of such two tuples is not possible for arbitrary downward closures of join-irreducibles. Fortunately, however, we do not require this property for arbitrary downward closures. In fact, our downward closures are rather special, i.e., possess an additional property which guarantees the ability to generate the desired tuples.

Lemma 12. *Let $N \in \mathcal{N}$ be a nested attribute and $\mathcal{X} \subset \mathcal{J}(N)$ be a proper subset of $\mathcal{J}(N)$ that is closed downwards with respect to $\leq$. Furthermore, $\mathcal{X}$ has the following property: for all $O \in N^{\oplus}$ with $O \in \mathcal{X}$ and all $Y, Z \in \mathcal{J}(N)$ that are branched with respect to O we have $Y \in \mathcal{X}$ or $Z \in \mathcal{X}$. Then there are distinct $t_1, t_2 \in dom(N)$ such that for all $X \in \mathcal{J}(N)$*

$$\pi^N_X(t_1) = \pi^N_X(t_2) \qquad \textit{if and only if} \qquad X \in \mathcal{X}$$

holds.

Proof. Lemma 13 deals with the case where $\mathcal{X}$ is empty. We can therefore assume that $\mathcal{X}$ is non-empty, i.e., a $\leq$-ideal. We proceed by induction on the structure of N. If $N = \lambda_L$ or $N = \lambda_A$ for some $A \in \mathcal{U}$, then $\mathcal{J}(N) = \emptyset$. Consequently, the hypotheses of the lemma are not met as $\mathcal{X}$ is supposed to be a proper subset of $\mathcal{J}(N)$. Hence, there is nothing to show.

If $N = A \in \mathcal{U}$, then there is nothing to show as $\mathcal{X} \neq \{A\}$ cannot occur since $\mathcal{X} \neq \mathcal{J}(N)$.

The case where $N = L[M]$ is a list-valued attribute is dealt with separately in Lemma 19.

Suppose $N = L(N_1, \ldots, N_k)$ is a record-valued attribute. We then know that

$$\mathcal{X} = \bigcup_{i=1}^{k} \{L(\lambda_{N_1}, \ldots, X_i, \ldots, \lambda_{N_k}) \mid X_i \in \mathcal{X}_i\}$$

for some $\leq$ downward closures $\mathcal{X}_i \subseteq \mathcal{J}(N_i)$ for all $i = 1, \ldots, k$. Suppose $O_i \in N_i^{\oplus}$, $O_i \in \mathcal{X}_i$ and $X_1^i, X_2^i \in \mathcal{J}(N_i)$ are branched with respect to O_i. Then

$$L(\lambda_{N_1}, \ldots, X_1^i, \ldots, \lambda_{N_k}) \text{ and } L(\lambda_{N_1}, \ldots, X_2^i, \ldots, \lambda_{N_k})$$

are branched with respect to $O = L(\lambda_{N_1}, \ldots, O_i, \ldots, \lambda_{N_k}) \in \mathcal{X}$. Consequently, $L(\lambda_{N_1}, \ldots, X_1^i, \ldots, \lambda_{N_k}) \in \mathcal{X}$ or $L(\lambda_{N_1}, \ldots, X_2^i, \ldots, \lambda_{N_k}) \in \mathcal{X}$, i.e., $X_1^i \in \mathcal{X}_i$ or $X_2^i \in \mathcal{X}_i$. Hence, the $\leq$ downward closures $\mathcal{X}_1, \ldots, \mathcal{X}_k$ satisfy the hypothesis of the lemma. We can conclude, by hypothesis, that for $i = 1, \ldots, k$ there are $t_1^i, t_2^i \in dom(N_i)$ such that for all $X_i \in \mathcal{J}(N_i)$ we have $\pi_{X_i}^{N_i}(t_1^i) = \pi_{X_i}^{N_i}(t_2^i)$ if and only if $X_i \in \mathcal{X}_i$. We now choose $t_1 := (t_1^1, \ldots, t_k^1) \in dom(N)$ and $t_2 := (t_1^2, \ldots, t_k^2) \in dom(N)$. For all $X \in \mathcal{J}(N)$ we then have $\pi_X^N(t_1) = \pi_X^N(t_2)$ if and only if $X \in \mathcal{X}$ holds.

Suppose $N = L(N_1 \oplus \cdots \oplus N_k)$ is a union-valued attribute. For $\mathcal{X} \neq \emptyset$ we then have

$$\mathcal{X} = \{L(\lambda_{N_1} \oplus \cdots \oplus \lambda_{N_k})\} \cup \bigcup_{i=1}^{k} \{L(\lambda_{N_1} \oplus \cdots \oplus X_i \oplus \cdots \oplus \lambda_{N_k}) \mid X_i \in \mathcal{X}_i\}$$

for some $\leq$ downward closures $\mathcal{X}_i \subseteq \mathcal{J}(N_i)$ for all $i = 1, \ldots, k$. For $O_i \in N_i^{\oplus}$ with $O \in \mathcal{X}_i$ and $X_1^i, X_2^i \in \mathcal{J}(N_i)$ that are branched with respect to O_i we have that

$$X_1 = L(\lambda_{N_1} \oplus \cdots \oplus X_1^i \oplus \cdots \oplus \lambda_{N_k}) \text{ and } X_2 = L(\lambda_{N_1} \oplus \cdots \oplus X_2^i \oplus \cdots \oplus \lambda_{N_k})$$

are branched with respect to $O = L(\lambda_{N_1} \oplus \cdots \oplus O_i \oplus \cdots \oplus \lambda_{N_k}) \in \mathcal{X}$. Consequently, $X_1 \in \mathcal{X}$ or $X_2 \in \mathcal{X}$, and therefore $X_1^i \in \mathcal{X}_i$ or $X_2^i \in \mathcal{X}_i$. Consequently, $\mathcal{X}_1, \ldots, \mathcal{X}_k$ satisfy the hypothesis of the lemma. We can conclude, by hypothesis, that for $i = 1, \ldots, k$ there are $t_1^i, t_2^i \in dom(N_i)$ such that for all $X_i \in \mathcal{J}(N_i)$ we have $\pi_{X_i}^{N_i}(t_1^i) = \pi_{X_i}^{N_i}(t_2^i)$ if and only if $X_i \in \mathcal{X}_i$. Since $L(\lambda_{N_1} \oplus \cdots \oplus \lambda_{N_k}) \in \mathcal{X}$ it follows from Definition 9 and the assumptions of the lemma that there is exactly one i with $1 \leq i \leq k$ such that $L(\lambda_{N_1} \oplus \cdots \oplus N_i \oplus \cdots \oplus \lambda_{N_k}) \notin \mathcal{X}$. We choose $t_1 := t_1^i \in dom(N_i)$ and $t_2 := t_2^i \in dom(N_i)$. It remains to show that for all $X \in \mathcal{J}(N)$ we have $\pi_X^N(t_1) = \pi_X^N(t_2)$ if and only if $X \in \mathcal{X}$ holds. For $X = L(\lambda_{N_1} \oplus \cdots \oplus \lambda_{N_k}) \in \mathcal{X}$ we have $\pi_X^N(t_1) = ok_i = \pi_X^N(t_2)$. Let $X = L(\lambda_{N_1} \oplus \cdots \oplus Y \oplus \cdots \oplus \lambda_{N_k}) \in \mathcal{J}(N)$. If $Y \not\leq N_i$, then $X \in \mathcal{X}$ and also $\pi_X^N(t_1) = ok_i = \pi_X^N(t_2)$. If $Y \leq N_i$, then $\pi_Y^{N_i}(t_1^i) = \pi_Y^{N_i}(t_2^i)$ if and only if $Y \in \mathcal{X}_i$. Consequently, $\pi_X^N(t_1) = \pi_X^N(t_2)$ if and only if $X \in \mathcal{X}$. □

The next lemma shows that it is always possible to find tuples that disagree on every join-irreducible. However, it depends on our assumption that the domain of every flat attribute contains at least two distinct elements.

Lemma 13. *Let $N \in \mathcal{N}$ be a nested attribute such that $\mathcal{J}(N) \neq \emptyset$. Then there are distinct $t_1, t_2 \in dom(N)$ such that for all $X \in \mathcal{J}(N)$ we have $\pi_X^N(t_1) \neq \pi_X^N(t_2)$.*

Proof. We proceed by structural induction on N. If $N = \lambda_L$ or $N = \lambda_A$ for some $A \in \mathcal{U}$, then $\mathcal{J}(N) = \emptyset$ and there is nothing to show. Suppose $N = A \in \mathcal{U}$.

Then choose $t_1 := a$ and $t_2 := a'$ with distinct values $a, a' \in dom(A)$ (recall that we assume that the domain of each flat attribute contains at least two distinct elements).

Let $N = L(N_1, \ldots, N_k)$ be a record-valued attribute with $\mathcal{J}(N) \neq \emptyset$. Then

$$\mathcal{J}(N) = \bigcup_{i=1}^{k} \{L(\lambda_{N_1}, \ldots, X_i, \ldots, \lambda_{N_k}) \mid X_i \in \mathcal{J}(N_i)\}.$$

It follows that there is some i such that $1 \leq i \leq k$ and $\mathcal{J}(N_i) \neq \emptyset$. The hypothesis tells us further that for all $i = 1, \ldots, k$ where $\mathcal{J}(N_i) \neq \emptyset$ there are distinct $t_1^i, t_2^i \in dom(N_i)$ such that for all $X_i \in \mathcal{J}(N_i)$ we have $\pi_{X_i}^{N_i}(t_1^i) \neq \pi_{X_i}^{N_i}(t_2^i)$. For those $i \in \{1, \ldots, k\}$ where $\mathcal{J}(N_i) = \emptyset$ holds let $t_1^i = t_2^i \in dom(N_i)$. We choose $t_1 := (t_1^1, \ldots, t_k^1) \in dom(N)$ and $t_2 := (t_1^2, \ldots, t_k^2) \in dom(N)$.

Let $N = L[M]$ be a list-valued attribute. We have that

$$\mathcal{J}(N) = \{L[\lambda_M]\} \cup \{L[M'] \mid M' \in \mathcal{J}(M)\}.$$

In this case we choose $t_1 := [\,]$ and $t_2 := [v]$ with some value $v \in dom(M)$. Note that $\pi_{L[\lambda_M]}^N(t_1) \neq \pi_{L[\lambda_M]}^N(t_2)$ and every element of $\mathcal{J}(N)$ is a superattribute of $L[\lambda_M]$. It remains to consider the case where $N = L(N_1 \oplus \cdots \oplus N_k)$ is a union-valued attribute. We then have

$$\mathcal{J}(N) = \{L(\lambda_{N_1} \oplus \cdots \oplus \lambda_{N_k})\} \cup \bigcup_{i=1}^{k} \{L(\lambda_{N_1} \oplus \cdots \oplus X_i \oplus \cdots \oplus \lambda_{N_k}) \mid X_i \in \mathcal{J}(N_i)\}.$$

Since $L(\lambda_{N_1} \oplus \cdots \oplus \lambda_{N_k})$ is subattribute of any other join-irreducible of N it suffices to find $t_1, t_2 \in dom(N)$ such that $\pi_{L(\lambda_{N_1} \oplus \cdots \oplus \lambda_{N_k})}^N(t_1) \neq \pi_{L(\lambda_{N_1} \oplus \cdots \oplus \lambda_{N_k})}^N(t_2)$. Since $k \geq 2$ we choose $t_1 \in dom(N_1)$ and $t_2 \in dom(N_2)$. □

3.7 List-valued attributes

In this section we will finalise the completeness proof of Theorem 11. In fact, it remains to prove Lemma 12 in case that we are dealing with a list-valued attribute. Given some $\leq$-ideal we will construct two lists that have matching projections on precisely those subattributes that belong to the ideal. The elements of the list are determined by i) the maximal join-irreducibles of the underlying nested attribute and ii) the maximal join-irreducibles that belong to the ideal.

The record constructor uses aggregation to enforce a more complex nesting structure. In other words, a complex nested data element derived by means of the record constructor is aggregated out of more basic data elements. The next definition captures these more basic elements, or *components*, on the attribute level.

Definition 14. Let $N \in \mathcal{N}$. The set $\mathcal{C}(N) \subseteq Sub(N)$ of *components* of N is inductively defined as follows

- $\mathcal{C}(\lambda_L) = \{\lambda_L\}$ for $L \in \mathcal{L}$, $\mathcal{C}(\lambda_A) = \{\lambda_A\}$ for $A \in \mathcal{U}$,
- $\mathcal{C}(A) = \{A\}$ for $A \in \mathcal{U}$,
- $\mathcal{C}(L(N_1, \ldots, N_k)) = \bigcup_{i=1}^{k} \{L(\lambda_{N_1}, \ldots, M_i, \ldots, \lambda_{N_k}) \mid M_i \in \mathcal{C}(N_i)\}$,
- $\mathcal{C}(L[N']) = \{L[M] \mid M \in \mathcal{C}(N')\}$, and
- $\mathcal{C}(L(N_1 \oplus \cdots \oplus N_k)) = \{L(N_1 \oplus \cdots \oplus N_k)\}$. □

Intuitively, an element of the domain of a nested attribute is an aggregation of elements from its component's domains.

Example 16. For $N = L(M(A \oplus K[O(B \oplus C)]), P(Q(D \oplus E \oplus F), G))$ the following three subattributes form the set of components of N:

- $L(M(A \oplus K[O(B \oplus C)]), P(\lambda_Q, \lambda_G))$,
- $L(\lambda_M, P(Q(D \oplus E \oplus F), \lambda_G))$,
- $L(\lambda_M, P(\lambda_Q, G))$. □

One may wonder why in Definition 14 union-valued attributes only have themselves as component (item five). This can be explained as follows. The presence of the union constructor leaves us with a choice between elements of different domains. For instance, a value from the domain of $Q(D \oplus E \oplus F)$ is a value from either the domain of D, or E or F. However, we cannot choose two different values at the same time. When defining the elements of the lists in our two-element nested database instance we need to make such a choice (any choice will do).

Let $Max\mathcal{J}(N)$ denote the set of those subattributes of N that are maximal in $\mathcal{J}(N) \cup \{\lambda_N\}$ with respect to $\leq$. Notice that $Max\mathcal{J}(N)$ is always non-empty, i.e., contains either all maximal join-irreducibles or λ_N if $\mathcal{J}(N) = \emptyset$.

Definition 15. A *choice* of a nested attribute N is a set $Ch(N) \subseteq Max\mathcal{J}(N)$ such that for all $C \in \mathcal{C}(N)$ there is precisely one $M \in Ch(N)$ with $M \leq C$. An M-choice of N is a pair $(M, Ch(N))$ where $M \in Ch(N)$ and $Ch(N)$ is a choice of N. □

Example 17. Consider the nested attribute N of Example 16. One possible choice of N consists of the following three maximal join-irreducibles of N:

- $L(M(\lambda_A \oplus K[O(B \oplus \lambda_C)]), P(\lambda_Q, \lambda_G))$,

– $L(\lambda_M, P(Q(D \oplus \lambda_E \oplus \lambda_F), \lambda_G))$,

– $L(\lambda_M, P(\lambda_Q, G))$. □

We will now define generic elements $\tau^N_{Ch(N)}(X)$ from the domain of a nested attribute N based on a subattribute $X \in Sub(N)$ and with respect to a choice $Ch(N)$. Suppose we are given some $\leq$-ideal, a maximal join-irreducible M and a subattribute Y that is $\leq$-maximal with the property that it is both a subattribute of M and a member of the ideal. The definition of $\tau^N_{Ch(N)}(X)$ is such that the projections of the elements $\tau^N_{Ch(N)}(Y)$ and $\tau^N_{Ch(N)}(M)$ coincide on precisely those subattributes that belong to ideal. Notice that the following definition is based on our assumption that the domain of each flat attribute contains at least two distinct values.

Definition 16. Let $N \in \mathcal{N}$, $Ch(N)$ a choice of N, and $X \in \mathcal{J}(N) \cup \{\lambda_N\}$. The identifying term $\tau^N_{Ch(N)}(X) \in dom(N)$ of X in N with respect to $Ch(N)$ is inductively defined as follows:

– $\tau^{\lambda_L}_{\{\lambda_L\}}(\lambda_L) = ok_L$, and $\tau^{\lambda_A}_{\{\lambda_A\}}(\lambda_A) = ok_A$,

– $\tau^A_{\{A\}}(\lambda_A) = a' \in dom(A)$ and $\tau^A_{\{A\}}(A) = a \in dom(A)$ where $a \neq a'$,

– if $N = L(N_1, \ldots, N_k)$ and $X = L(X_1, \ldots, X_k)$, then

$$\tau^N_{Ch(N)}(X) = (\tau^{N_1}_{Ch(N_1)}(X_1), \ldots, \tau^{N_k}_{Ch(N_k)}(X_k)),$$

– if $N = L[N']$, then

$$\tau^N_{Ch(N)}(X) = \begin{cases} [\tau^{N'}_{Ch(N')}(X')] & \text{, if } X = L[X'] \\ [\,] & \text{, if } X = \lambda_L \end{cases}$$

– if $N = L(\lambda_{N_1} \oplus \cdots \oplus \lambda_{N_k})$, then

$$\tau^N_{\{N\}}(X) = \begin{cases} \tau^{\lambda_{N_1}}_{\{\lambda_{N_1}\}}(\lambda_{N_1}) & \text{, if } X = L(\lambda_{N_1} \oplus \cdots \oplus \lambda_{N_k}) \\ \tau^{\lambda_{N_2}}_{\{\lambda_{N_2}\}}(\lambda_{N_2}) & \text{, if } X = \lambda_L \end{cases}$$

– if $N = L(N_1 \oplus \cdots \oplus N_k)$, $M = L(\lambda_{N_1} \oplus \cdots M_i \cdots \oplus \lambda_{N_k})$ and $M_j \in Max\mathcal{J}(N_j)$ such that $j \neq i$, then

$$\tau^N_{\{M\}}(X) = \begin{cases} \tau^{N_i}_{\{M_i\}}(X_i) & \text{, if } X = L(X_1 \oplus \cdots \oplus X_k) \\ \tau^{N_j}_{\{M_j\}}(\lambda_{N_j}) & \text{, if } X = \lambda_L \end{cases}$$

□

In order to emphasise that $M \in Max\mathcal{J}(N)$ is part of a choice $Ch(N)$ of N we denote the identifying term $\tau^N_{Ch(N)}(X)$ by $\tau^N_{(M,Ch(N))}(X)$.

The next lemma establishes the first property of the identifying terms. Given some $\leq$-ideal, a maximal join-irreducible M and a subattribute $Y \leq M$ that is maximal with this property in the ideal, then the identifying terms of M and Y coincide on projections to any elements of the ideal.

Lemma 17. *Let $N \in \mathcal{N}$ be a nested attribute, $\mathcal{Y} \subset \mathcal{J}(N) \cup \{\lambda_N\}$ some $\leq$-ideal, $M \in Max\mathcal{J}(N)$ and $Y \in \{Z \in \mathcal{Y} \mid Z \leq M$ and $\forall Z' \in \mathcal{Y}.(Z \leq Z'$ and $Z' \leq M) \Rightarrow Z' = Z\}$. Then for all $Z \in \mathcal{Y}_{max} = \{Z \in \mathcal{Y} \mid \forall Z' \in \mathcal{Y}.Z \leq Z' \Rightarrow Z' = Z\}$*

$$\pi^N_Z(\tau^N_{(M,Ch(N))}(M)) = \pi^N_Z(\tau^N_{(M,Ch(N))}(Y)) \tag{3.1}$$

holds.

Proof. If $M = \lambda_N$, then $Y = \lambda_N$ and (3.1) is true. Moreover, if $\mathcal{Y} = \{\lambda_N\}$, then $\mathcal{Y}_{\max} = \{\lambda_N\}$ and (3.1) is true. We will therefore assume for the remainder of the proof that $M \neq \lambda_N$ and $\mathcal{Y} \neq \{\lambda_N\}$.

We show (3.1) by induction on the structure of N.

If $N = \lambda_L$ or $N = \lambda_A$ for some $L \in \mathcal{L}$ and $A \in \mathcal{U}$, then $M = \lambda_N$. In case that $N = A \in \mathcal{U}$, then all possible cases have already been covered.

Consider the case where $N = L(N_1, \ldots, N_k)$. For $i = 1, \ldots, k$ let

$$\mathcal{Y}_i = \{W_i \in \mathcal{J}(N_i) \cup \{\lambda_{N_i}\} \mid L(\lambda_{N_1}, \ldots, W_i, \ldots, \lambda_{N_k}) \in \mathcal{Y}\}.$$

Let $M = L(\lambda_{N_1}, \ldots, M_i, \ldots, \lambda_{N_k})$ such that $M_i \in Max\mathcal{J}(N_i)$, and let $Y = L(\lambda_{N_1}, \ldots, Y_i, \ldots, \lambda_{N_k})$ with $Y_i \in \{Z_i \in \mathcal{Y}_i \mid Z_i \leq M_i$ and $\forall Z_i' \in \mathcal{Y}_i.(Z_i \leq Z_i'$ and $Z_i' \leq M_i) \Rightarrow Z_i' = Z_i\}$. Moreover, for all $i = 1, \ldots, k$ let

$$\mathcal{Y}^i_{\max} = \{W_i \in \mathcal{J}(N_i) \cup \{\lambda_{N_i} \mid L(\lambda_{N_1}, \ldots, W_i, \ldots, \lambda_{N_k}) \in \mathcal{Y}_{\max}\},$$

an antichain with respect to $\leq$.

Consider first the case where $Z = L(\lambda_{N_1}, \ldots, Z_i, \ldots, \lambda_{N_k})$ with $Z_i \in \mathcal{Y}^i_{\max}$. Induction hypothesis shows then that

$$\pi^{N_i}_{Z_i}(\tau^{N_i}_{(M_i,Ch(N_i))}(M_i)) = \pi^{N_i}_{Z_i}(\tau^{N_i}_{(M_i,Ch(N_i))}(Y_i))$$

holds, and therefore $\pi^N_Z(\tau^N_{(M,Ch(N))}(M)) = \pi^N_Z(\tau^N_{(M,Ch(N))}(Y))$.

Consider now the case where $Z = L(\lambda_{N_1}, \ldots, Z_j, \ldots, \lambda_{N_k})$ with $Z_j \in \mathcal{Y}^j_{\max}$ and $j \neq i$. Then also $\pi^N_Z(\tau^N_{(M,Ch(N))}(M)) = \pi^N_Z(\tau^N_{(M,Ch(N))}(Y))$ since

$$\pi^{N_i}_{\lambda_{N_i}}(\tau^{N_i}_{(M_i,Ch(N_i))}(M_i)) = \pi^{N_i}_{\lambda_{N_i}}(\tau^{N_i}_{(M_i,Ch(N_i))}(Y_i))$$

and

$$\pi^{N_j}_{Z_j}(\tau^{N_j}_{Ch(N_j)}(\lambda_{N_j})) = \pi^{N_j}_{Z_j}(\tau^{N_j}_{Ch(N_j)}(\lambda_{N_j}))$$

and for $m \in \{1, \ldots, k\}$ with $m \neq i$ and $m \neq j$ we have

$$\pi^{N_m}_{\lambda_{N_m}}(\tau^{N_m}_{Ch(N_m)}(\lambda_{N_m})) = \pi^{N_m}_{\lambda_{N_m}}(\tau^{N_m}_{Ch(N_m)}(\lambda_{N_m})).$$

This shows that (3.1) holds in this case.

The cases where N is a list-valued attribute and a union-valued attribute, respectively, can be shown in a similar way. □

The next lemma establishes the second property of the identifying terms. Given some $\leq$-ideal, a maximal join-irreducible M and a subattribute $Y \leq M$ that is maximal with this property in the ideal, then the identifying terms of M and Y differ on projections to any subattribute X where $Y < X \leq M$ holds.

Lemma 18. *Let $N \in \mathcal{N}$ be a nested attribute, $\mathcal{Y} \subset \mathcal{J}(N) \cup \{\lambda_N\}$ some $\leq$-ideal, $M \in Max\mathcal{J}(N)$ and $Y \in \{Z \in \mathcal{Y} \mid Z \leq M$ and $\forall Z' \in \mathcal{Y}.(Z \leq Z'$ and $Z' \leq M) \Rightarrow Z' = Z\}$. For all $X \in \mathcal{J}(N)$ where $Y < X \leq M$ we have*

$$\pi^N_X(\tau^N_{(M,Ch(N))}(M)) \neq \pi^N_X(\tau^N_{(M,Ch(N))}(Y)) \quad . \tag{3.2}$$

Proof. If $M = \lambda_N$, then $Y = \lambda_N$ and (3.2) is true as there is no $X \in \mathcal{J}(N)$ such that $Y < X \leq M$ holds. For the remainder of the proof we therefore assume that $M \neq \lambda_N$.

We proceed by induction on the structure of N. There is nothing to show for $N = \lambda_L$ and $N = \lambda_A$ for some $L \in \mathcal{L}$ and $A \in \mathcal{U}$. In case that $N = A \in \mathcal{U}$ the only case to consider is where $M = A$ and $\mathcal{Y} = \{Y\}$ with $Y = \lambda_A$, and $X = A$. However, in this case we have

$$\pi^A_A(\tau^A_{(A,\{A\})}(A)) = a \neq a' = \pi^A_A(\tau^A_{(A,\{A\})}(\lambda_A)).$$

Consider now the case where $N = L(N_1, \ldots, N_k)$. For $i = 1, \ldots, k$ let

$$\mathcal{Y}_i = \{W_i \in \mathcal{J}(N_i) \cup \{\lambda_{N_i}\} \mid L(\lambda_{N_1}, \ldots, W_i, \ldots, \lambda_{N_k}) \in \mathcal{Y}\}.$$

Let $M = L(\lambda_{N_1}, \ldots, M_i, \ldots, \lambda_{N_k})$ such that $M_i \in Max\mathcal{J}(N_i)$, and let $Y = L(\lambda_{N_1}, \ldots, Y_i, \ldots, \lambda_{N_k})$ with $Y_i \in \{Z_i \in \mathcal{Y}_i \mid Z_i \leq M_i$ and $\forall Z_i' \in \mathcal{Y}_i.(Z_i \leq Z_i'$ and $Z_i' \leq M_i) \Rightarrow Z_i' = Z_i\}$. Moreover, for all $i = 1, \ldots, k$ let

$$\mathcal{Y}^i_{\max} = \{W_i \in \mathcal{J}(N_i) \cup \{\lambda_{N_i} \mid L(\lambda_{N_1}, \ldots, W_i, \ldots, \lambda_{N_k}) \in \mathcal{Y}_{\max}\},$$

an antichain with respect to $\leq$. Let $X \in \mathcal{J}(N)$ such that $Y < X \leq M$ holds. We conclude that $X = L(\lambda_{N_1}, \ldots, X_i, \ldots, \lambda_{N_k})$ where $Y_i < X_i \leq M_i$ holds. In particular, since $(M, Ch(N))$ is an M-choice of N we know that $(M_i, Ch(N_i))$ is an M_i-choice of N_i. We conclude by hypothesis that

$$\pi^{N_i}_{X_i}(\tau^{N_i}_{(M_i,Ch(N_i))}(M_i)) \neq \pi^{N_i}_{X_i}(\tau^{N_i}_{(M_i,Ch(N_i))}(Y_i))$$

holds. For $Ch(N) = \bigcup_{i=1}^{k} \{L(\lambda_{N_1}, \ldots, M_i, \ldots, \lambda_{N_k}) : M_i \in Ch(N_i)\}$ we therefore know that

$$(\pi^{N_1}_{\lambda_{N_1}}(\tau^{N_1}_{Ch(N_1)}(\lambda_{N_1})), \ldots, \pi^{N_i}_{X_i}(\tau^{N_i}_{(M_i,Ch(N_i))}(M_i)), \ldots, \pi^{N_k}_{\lambda_{N_k}}(\tau^{N_k}_{Ch(N_k)}(\lambda_{N_k})))$$

and

$$(\pi^{N_1}_{\lambda_{N_1}}(\tau^{N_1}_{Ch(N_1)}(\lambda_{N_1})), \ldots, \pi^{N_i}_{X_i}(\tau^{N_i}_{(M_i,Ch(N_i))}(Y_i)), \ldots, \pi^{N_k}_{\lambda_{N_k}}(\tau^{N_k}_{Ch(N_k)}(\lambda_{N_k})))$$

are different. That is,

$$\pi^N_X(\tau^N_{(M,Ch(N))}(M)) \neq \pi^N_X(\tau^N_{(M,Ch(N))}(Y)).$$

Suppose $N = L[N']$. Let $M = L[M']$ with $M' \in Max\mathcal{J}(N')$. Let $\mathcal{Y} = \{\lambda_L\}$. Then $Y = \lambda_L$. Let $X \in \mathcal{J}(N)$ such that $Y < X \leq M$ holds. We conclude that $X = L[X']$ where $X' \leq M'$ holds. However, for $Y = \lambda_L$ we have

$$\pi^N_X(\tau^N_{(M,Ch(N))}(M)) = [\pi^{N'}_{X'}(\tau^{N'}_{(M',Ch(N'))}(M'))] \neq [\,] = \pi^N_X(\tau^N_{(M,Ch(N))}(Y)).$$

Let now $\mathcal{Y} \neq \{\lambda_L\}$. That is, $\mathcal{Y}' = \{W' \in \mathcal{J}(N') \cup \{\lambda_{N'}\} \mid L[W'] \in \mathcal{Y}\} \neq \emptyset$. Let $Y' \in \{Z \in \mathcal{Y}' \mid Z \leq M' \text{ and } \forall Z' \in \mathcal{Y}'.(Z \leq Z' \text{ and } Z' \leq M') \Rightarrow Z' = Z\}$. Let $X \in \mathcal{J}(N)$ such that $Y < X \leq M$ holds. We conclude that $X = L[X']$ where $Y' < X' \leq M'$ holds. Consequently,

$$\begin{aligned}\pi^N_X(\tau^N_{(M,Ch(N))}(M)) &= [\pi^{N'}_{X'}(\tau^{N'}_{(M',Ch(N'))}(M'))] \\ &\neq [\pi^{N'}_{X'}(\tau^{N'}_{(M',Ch(N'))}(Y'))] \\ &= \pi^N_X(\tau^N_{(M,Ch(N))}(Y)).\end{aligned}$$

Consider the case where $N = L(N_1 \oplus \cdots \oplus N_k)$. For $i = 1, \ldots, k$ let $\mathcal{Y}_i = \{W_i \in \mathcal{J}(N_i) \cup \{\lambda_{N_i}\} \mid L(\lambda_{N_1} \oplus \cdots W_i \cdots \oplus \lambda_{N_k}) \in \mathcal{Y}\}$. According to our assumption that $M > \lambda_L$ we have that $M = L(\lambda_{N_1} \oplus \cdots M_i \cdots \oplus \lambda_{N_k})$ where $M_i \in Max\mathcal{J}(N_i)$ holds. Let $M_i > \lambda_{N_i}$. Let $X = L(\lambda_{N_1} \oplus \cdots X_i \cdots \oplus \lambda_{N_k}) \in \mathcal{J}(N)$ such that $Y < X \leq M$ holds. If $Y = \lambda_L$, then for some $M_j \in Ch(N_j)$ such that $j \neq i$ we have

$$\begin{aligned}\pi^N_X(\tau^N_{(M,Ch(N))}(M)) &= \pi^{N_i}_{X_i}(\tau^{N_i}_{(M_i,Ch(N_i))}(M_i)) \\ &\neq \pi^{N_j}_{\lambda_{N_j}}(\tau^{N_j}_{Ch(N_j)}(\lambda_{N_j})) \\ &= \pi^N_X(\tau^N_{(M,Ch(N))}(Y)).\end{aligned}$$

For $Y = L(\lambda_{N_1} \oplus \cdots Y_i \cdots \oplus \lambda_{N_k})$ with $Y_i \in \mathcal{Y}_i$ and $Y_i < X_i \leq M_i$ we apply the induction hypothesis to conclude that

$$\begin{aligned}\pi^N_X(\tau^N_{(M,Ch(N))}(M)) &= \pi^{N_i}_{X_i}(\tau^{N_i}_{(M_i,Ch(N_i))}(M_i)) \\ &\neq \pi^{N_i}_{X_i}(\tau^{N_i}_{(M_i,Ch(N_i))}(Y_i)) \\ &= \pi^N_X(\tau^N_{(M,Ch(N))}(Y)).\end{aligned}$$

It remains to consider the case where $N_i = \lambda_{N_i}$ for all $i = 1, \ldots, k$. Then $N = M = X = L(\lambda_{N_1} \oplus \cdots \oplus \lambda_{N_k})$ and $Y = \lambda_L$. We then have

$$\begin{aligned}\pi_X^N(\tau^N_{(M,Ch(N))}(M)) &= \tau^{\lambda_{N_1}}_{\{\lambda_{N_1}\}}(\lambda_{N_1}) \\ &\neq \tau^{\lambda_{N_2}}_{\{\lambda_{N_2}\}}(\lambda_{N_2}) \\ &= \pi_X^N(\tau^N_{(M,Ch(N))}(Y)).\end{aligned}$$

This concludes the proof of (3.2). □

We will now combine Lemmata 17 and 18 to finalise our construction of the desired two-element nested database instance by providing the construction for list-valued attributes.

Lemma 19. *Let $L[N] \in \mathcal{N}$ be a list valued attribute and $\emptyset \neq \mathcal{X} \subset \mathcal{J}(L[N])$ some $\leq$-ideal. Then there are distinct $t_1, t_2 \in dom(L[N])$ such that for all $X \in \mathcal{J}(L[N])$ we have*

$$\pi_X^{L[N]}(t_1) = \pi_X^{L[N]}(t_2) \qquad \textit{if and only if} \qquad X \in \mathcal{X}.$$

Proof. Since $\mathcal{X} \neq \emptyset$ we have $\mathcal{X} = \{L[Y] \mid Y \in \mathcal{Y}\}$ for some $\leq$-ideal $\mathcal{Y} \subseteq \mathcal{J}(N) \cup \{\lambda_N\}$ with $\lambda_N \in \mathcal{Y}$. Since $\mathcal{X} \neq \mathcal{J}(L[N])$ we also have $\mathcal{Y} \neq \mathcal{J}(N)$. Let $\{M_1, \ldots, M_l\} = \{M \in Max\mathcal{J}(N) \mid M \notin \mathcal{Y}\}$. For $i = 1, \ldots, l$ let $Y_i \in \mathcal{Y}$ such that $Y_i \leq M_i$ and for all $Z \in \mathcal{Y}$ with $Z \leq M_i$ and $Y_i \leq Z$ follows $Z = Y_i$. For $i = 1, \ldots, l$ let $(M_i, Ch_i(N))$ be an M_i-choice of N. Define $t_1, t_2 \in dom(L[N])$ by

$$\begin{aligned}t_1 &:= [\tau^N_{(M_1,Ch_1(N))}(M_1), \ldots, \tau^N_{(M_l,Ch_l(N))}(M_l)], \text{ and} \\ t_2 &:= [\tau^N_{(M_1,Ch_1(N))}(Y_1), \ldots, \tau^N_{(M_l,Ch_l(N))}(Y_l)] \quad .\end{aligned}$$

For $X \in \mathcal{J}(L[N])$ we have $X = L[X']$ with $X' \in \mathcal{J}(N) \cup \{\lambda_N\}$. By definition of t_1, t_2 the statement of this lemma, i.e.,

$$\pi_X^{L[N]}(t_1) = \pi_X^{L[N]}(t_2) \quad \text{if and only if} \quad X \in \mathcal{X}$$

becomes equivalent to

$$\pi^N_{X'}(\tau^N_{(M_i,Ch_i(N))}(M_i)) = \pi^N_{X'}(\tau^N_{(M_i,Ch_i(N))}(Y_i)) \forall i = 1, \ldots, l \text{ iff } X' \in \mathcal{Y}.$$

Note that $\pi^{L[N]}_{L[\lambda_N]}(t_1) = \pi^{L[N]}_{L[\lambda_N]}(t_2)$ holds. Lemma 17 demonstrates that for all $i = 1, \ldots, l$ and for all $\leq$-maximal elements $X' \in \mathcal{Y}$ we have

$$\pi^N_{X'}(\tau^N_{(M_i,Ch_i(N))}(M_i)) = \pi^N_{X'}(\tau^N_{(M_i,Ch_i(N))}(Y_i)).$$

Moreover, for each $X' \in \mathcal{J}(N) - \mathcal{Y}$ there is some $i \in \{1, \ldots, l\}$ such that $Y_i < X' \leq M_i$ holds. Lemma 18 demonstrates that

$$\pi^N_{X'}(\tau^N_{(M_i,Ch_i(N))}(M_i)) \neq \pi^N_{X'}(\tau^N_{(M_i,Ch_i(N))}(Y_i))$$

holds. This completes the proof. □

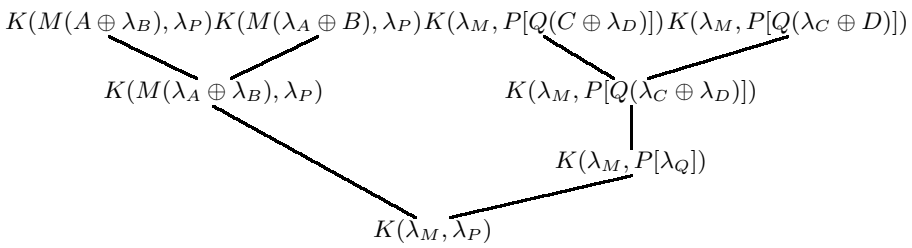

Figure 4: Structure of Join Irreducibles in Example 18

The description of generating the desired two-element relation is now complete. This finalises the proof of Theorem 11. The following example illustrates the construction described in the proof of Lemma 19.

Example 18. Consider the list-valued attribute $L[N]$ where

$$N = K(M(A \oplus B), P[Q(C \oplus D)]).$$

The structure of $\mathcal{J}(N) \cup \{\lambda_N\}$ is illustrated in Figure 4. Suppose the $\leq$-ideal $\mathcal{X}$ on $L[N]$ is

$$\{\ L[K(M(\lambda_A \oplus \lambda_B), \lambda_P)],\ L[K(\lambda_M, P[\lambda_Q])],\ L[K(\lambda_M, \lambda_P)]\ \}$$

and therefore the corresponding $\leq$-ideal $\mathcal{Y} \subseteq \mathcal{J}(N) \cup \{\lambda_N\}$ is

$$\mathcal{Y} = \{\ K(M(\lambda_A \oplus \lambda_B), \lambda_P),\ K(\lambda_M, P[\lambda_Q]),\ K(\lambda_M, \lambda_P)\ \}.$$

None of the maximal join-irreducibles of N are in $\mathcal{Y}$, i.e., we have

- $M_1 = K(M(A \oplus \lambda_B), \lambda_P)$ and $Y_1 = K(M(\lambda_A \oplus \lambda_B), \lambda_P)$,
- $M_2 = K(M(\lambda_A \oplus B), \lambda_P)$ and $Y_2 = K(M(\lambda_A \oplus \lambda_B), \lambda_P)$,
- $M_3 = K(\lambda_M, P[Q(C \oplus \lambda_D)])$ and $Y_3 = K(\lambda_M, P[\lambda_Q])$,
- $M_4 = K(\lambda_M, P[Q(\lambda_C \oplus D)])$ and $Y_4 = K(\lambda_M, P[\lambda_Q])$.

According to the proof of Lemma 19 we form the following identifying terms with respect to some choices of N:

- $\tau^N_{(M_1,\{M_1,M_3\})}(M_1) = (\tau^{M(A\oplus B)}_{\{M(A\oplus\lambda_B)\}}(M(A \oplus \lambda_B)), \tau^{P[Q(C\oplus D)]}_{\{P[Q(C\oplus\lambda_D)])\}}(\lambda_P)) = (a, [\,])$,
- $\tau^N_{(M_1,\{M_1,M_3\})}(Y_1) = (\tau^{M(A\oplus B)}_{\{M(A\oplus\lambda_B)\}}(M(\lambda_A \oplus \lambda_B)), \tau^{P[Q(C\oplus D)]}_{\{P[Q(C\oplus\lambda_D)])\}}(\lambda_P)) = (a', [\,])$,

– $\tau^N_{(M_2,\{M_2,M_3\})}(M_2) = (b, [\,])$ and $\tau^N_{(M_2,\{M_2,M_3\})}(Y_2) = (b', [\,])$,

– $\tau^N_{(M_3,\{M_2,M_3\})}(M_3) = (a', [c])$ and $\tau^N_{(M_3,\{M_2,M_3\})}(Y_3) = (a', [d'])$,

– $\tau^N_{(M_4,\{M_1,M_4\})}(M_4) = (b', [d])$ and $\tau^N_{(M_4,\{M_1,M_4\})}(Y_4) = (b', [c'])$.

We therefore obtain the following two lists

$$t_1 = [(a, [\,]), (b, [\,]), (a', [c]), (b', [d])]$$
$$t_2 = [(a', [\,]), (b', [\,]), (a', [d']), (b', [c'])]$$

and for all $X \in \mathcal{J}(N)$ we have $\pi^{L[N]}_X(t_1) = \pi^{L[N]}_X(t_2)$ if and only if $X \in \mathcal{X}$. □

Example 19. Let $N = K(M(A \oplus B), P[Q(C \oplus D)])$ just as in the previous example. Hence, the structure of the join-irreducibles is illustrated in Figure 4. Suppose

$$\Sigma = \{K(\lambda_M, P[\lambda_Q]) \to_w K(M(\lambda_A \oplus \lambda_B), \lambda_P)\}.$$

It is relatively easy to see that the wFD

$$\varphi_1 = K(\lambda_M, P[Q(C \oplus \lambda_D)]) \to_w K(M(A \oplus \lambda_B), \lambda_P)$$

is not implied by Σ. In fact, the two tuples

$$t_1 = (a, [c]) \quad \text{and} \quad t_2 = (a', [c])$$

with distinct elements $a, a' \in dom(A)$ form a counterexample for the implication of φ_1 by Σ. However, the wFD

$$\varphi_2 = K(\lambda_M, P[Q(C \oplus \lambda_D)]) \to_w \{K(M(A \oplus \lambda_B), \lambda_P), K(M(\lambda_A \oplus B), \lambda_P)\}$$

is indeed implied by Σ. In fact, we will now show an inference of φ_2 from Σ. In a fist step we infer the wFD

$$\{K(\lambda_M, P[\lambda_Q]), K(\lambda_M, P[Q(\lambda_C \oplus \lambda_D)]), K(\lambda_M, P[Q(C \oplus \lambda_D)])\} \to_w K(M(\lambda_A \oplus \lambda_B), \lambda_P)\}$$

from Σ by an application of *Rule 2*. Subsequently, we apply *Rule 3* to infer

$$K(\lambda_M, P[Q(C \oplus \lambda_D)]) \to_w K(M(\lambda_A \oplus \lambda_B), \lambda_P).$$

Moreover, we apply *Rule 1* to infer

$$K(\lambda_M, P[Q(C \oplus \lambda_D)]) \to_w$$
$$\{K(M(\lambda_A \oplus \lambda_B), \lambda_P), K(M(A \oplus \lambda_B), \lambda_P), K(M(\lambda_A \oplus B), \lambda_P)\}. \quad (3.3)$$

On the other hand, we can apply *Axiom 3* to infer

$$K(M(\lambda_A \oplus \lambda_B), \lambda_P) \to_w \{K(M(A \oplus \lambda_B), \lambda_P), K(M(\lambda_A \oplus B), \lambda_P)\}$$

and then *Rule 2* to obtain

$$\{K(M(\lambda_A \oplus \lambda_B), \lambda_P), K(\lambda_M, P[Q(C \oplus \lambda_D)])\} \rightarrow_w \{K(M(A \oplus \lambda_B), \lambda_P), K(M(\lambda_A \oplus B), \lambda_P)\}. \quad (3.4)$$

Finally, we apply *Rule 5* to (3.3) and (3.4) to infer φ_2 where $\mathcal{U} = \{K(M(\lambda_A \oplus \lambda_B), \lambda_P)\}$. Since every rule is sound, φ_2 is indeed implied by Σ. □

4 Logical Characterisation

In this section we will characterise the implication of weak functional dependencies in logical terms. Therefore, we will first define a mapping of wFDs to propositional clauses. Subsequently, we encode the non-trivial structure of the join-irreducibles as clauses as well. Finally, we also encode those wFDs that can be inferred by *Axiom 3* as clauses, too. This enables us to prove the desired equivalence. A database designer can take advantage of these equivalences to reduce database design problems to well-studied problems in Boolean propositional logic. For instance, state-of-the-art SAT solvers [Hirsch and Kojevnikov, 2005; Prasad et al., 2005] may be applied to reason about wFDs. Finally, the equivalences further show that relational database design solutions can be reused to solve problems for nested databases.

4.1 Weak Functional Dependencies and Clauses

Let $\phi : \mathcal{J}(N) \rightarrow \mathcal{V}$ denote a bijection between the join-irreducibles of N and the set $\mathcal{V}$ of Boolean propositional variables. We will now extend this bijection to weak functional dependencies over the nested attribute N and Boolean propositional clauses over $\mathcal{V}$.

It follows from the soundness of the inference rules *Rule 1*, *Rule 2*, *Rule 3*, and *Rule 4* that an instance $r \subseteq dom(N)$ satisfies the wFD $\mathcal{X} \rightarrow_w \mathcal{Y}$ if and only if r satisfies the wFD $\max(\mathcal{X}) \rightarrow_w \min(\mathcal{Y})$ where $\max(\mathcal{X}) = \{X \in \mathcal{X} \mid \forall Z \in \mathcal{X}.X \leq Z \Rightarrow X = Z\}$ and $\min(\mathcal{Y}) = \{Y \in \mathcal{Y} \mid \forall Z \in \mathcal{Y}.Z \leq Y \Rightarrow Y = Z\}$. Indeed, $\mathcal{X} \rightarrow_w \mathcal{Y}$ can be inferred from $\max(\mathcal{X}) \rightarrow_w \min(\mathcal{Y})$ by an application of *Rule 1* and an application of *Rule 2*. Vice versa, $\max(\mathcal{X}) \rightarrow_w \min(\mathcal{Y})$ can be inferred from $\mathcal{X} \rightarrow_w \mathcal{Y}$ by an application of *Rule 3* and an application of *Rule 4*. Without loss of generality, we assume for the remainder of this section that every wFD $\mathcal{X} \rightarrow_w \mathcal{Y}$ is of the form $\max(\mathcal{X}) \rightarrow_w \min(\mathcal{Y})$.

Consider the weak functional dependency $\varphi : \{X_1, \ldots, X_n\} \rightarrow_w \{Y_1, \ldots, Y_k\}$ on N. Define $\Phi(\varphi)$ to be the propositional formula

$$\varphi' = \neg\phi(X_1) \vee \cdots \vee \neg\phi(X_n) \vee \phi(Y_1) \vee \cdots \vee \phi(Y_k).$$

If Σ is a set of wFDs on N, then let $\Sigma' = \{\sigma' \mid \sigma \in \Sigma\}$ denote the corresponding set of propositional formulae over $\mathcal{V}$. Furthermore, if $\{J_1, \ldots, J_m\}$ denotes the set of maximal join-irreducibles of N, then let $\phi_N = \neg\phi(J_1) \vee \cdots \vee \neg\phi(J_m)$. The formula ϕ_N is the Boolean equivalent for the set semantics of database relations in which the same tuple cannot occur more than once.

Furthermore, the set

$$\Sigma_N^{\mathcal{J}} = \{\neg\phi(U) \vee \phi(V) \mid U, V \in \mathcal{J}(N), U \text{ covers}^1 V\}$$

denotes those clauses which encode the structure of (the join-irreducibles of) N. Finally, the set

$$\Sigma_N^{\mathcal{B}} = \{\neg\phi(O) \vee \phi(X) \vee \phi(Y) \mid O \in N^{\oplus}, X, Y \text{ branched with respect to } O\}$$

denotes those clauses which represent the trivial weak functional dependencies from *Axiom 3*.

4.2 The Equivalence

We will now present the main results of this paper. They generalise results from the relational data model [Fagin, 1977a; Sagiv et al., 1981; Sagiv et al., 1987] where

1. only nested attributes are considered that result from a single application of the record constructor to a finite number of flat attributes,
2. and the join-irreducibles of these nested attributes form an anti-chain.

In order to capture the implication of weak functional dependencies we require the propositional formula ϕ_N. This is already the case in the relational model of data [Sagiv et al., 1987]. As in the presence of record, list, set and multiset constructor [Hartmann and Link, 2008] we also require the formulae in $\Sigma_N^{\mathcal{J}}$. However, in the presence of record, list and disjoint-union constructor we also require the formulae in $\Sigma_N^{\mathcal{B}}$.

Theorem 20. *[Equivalence Theorem for Weak Functional Dependencies] Let $N \in \mathcal{N}$ be a nested attribute, and $\Sigma \cup \{\varphi\}$ a set of weak functional dependencies on N. Let Σ' denote the corresponding set of propositional formulae for Σ. Then*

1. *Σ implies φ,*
2. *Σ implies φ in the world of two-element relations, and*

[1] U covers V iff $U < V$ and for all $W \in \mathcal{J}(N)$ with $U \leq W \leq V$ we have $U = W$ or $V = W$, this is just the standard definition of a *cover relation* for posets, see [Anderson, 1987]

3. $\Sigma' \cup \Sigma_N^{\mathcal{J}} \cup \Sigma_N^{\mathcal{B}} \cup \{\phi_N\}$ *logically implies* φ'

are equivalent. □

We will now prove Theorem 20. We start off by showing the equivalence of *1)* and *2)* in Theorem 20. It is immediate that *1)* implies *2)* since every two-element relation is also a relation. The converse implication follows from Definition 6. Assume that *1)* does not hold. That is, there is some $r \subseteq dom(N)$ such that $\models_r \sigma$ for all $\sigma \in \Sigma$, but not $\models_r \varphi$. According to Definition 6 there must be some distinct $t_1, t_2 \in r$ such that not $\models_{\{t_1,t_2\}} \varphi$ holds. Since $\{t_1, t_2\} \subseteq r$ we must have $\models_{\{t_1,t_2\}} \sigma$ for all $\sigma \in \Sigma$. Consequently, Σ does not imply φ in the world of two-element relations, i.e., not *2)*.

In order to complete the proof of Theorem 20 it remains to show the equivalence between *2)* and *3)*. The key idea is to define truth assignments based on two-element relations and vice versa. In fact, one interprets a variable as *true* precisely if the two nested data elements coincide on their projections to the corresponding extended join-irreducible of that variable.

Lemma 21. *Let* φ *be a weak functional dependency on the nested attribute* N*, and* $r = \{t_1, t_2\} \subseteq dom(N)$ *such that* $t_1 \neq t_2$*. Then* $\models_r \varphi$ *if and only if* $\models_{\theta_r} \varphi'$ *where*

$$\theta_r(V) = \begin{cases} \text{true}, & \text{if } \pi^N_{\phi^{-1}(V)}(t_1) = \pi^N_{\phi^{-1}(V)}(t_2) \\ \text{false}, & \text{else} \end{cases}$$

for all $V \in \phi(\mathcal{J}(N))$*.*

Proof. Let φ denote the weak functional dependency

$$\{X_1, \ldots, X_n\} \to \{Y_1, \ldots, Y_k\}$$

on N. That is, φ' denotes the clause $\neg\phi(X_1) \vee \cdots \vee \neg\phi(X_n) \vee \phi(Y_1) \vee \cdots \vee \phi(Y_k)$.

We show the *if*-part first. Suppose θ_r makes φ' *true*. This means, θ_r must make at least one of $\phi(X_1), \ldots, \phi(X_n)$ *false* or at least one of $\phi(Y_1), \ldots, \phi(Y_k)$ *true*. If $\theta(\phi(X_i)) =$ *false*, then $\pi^N_{X_i}(t_1) \neq \pi^N_{X_i}(t_2)$ and r satisfies φ. If $\theta(\phi(Y_j)) =$ *true*, then $\pi^N_{Y_j}(t_1) \neq \pi^N_{Y_j}(t_2)$ and r satisfies φ again.

It remains to show the *only if*-part. Suppose r satisfies φ, i.e., $\pi^N_{X_i}(t_1) \neq \pi^N_{X_i}(t_2)$ for some $i \in \{1, \ldots, n\}$ or $\pi^N_{Y_j}(t_1) = \pi^N_{Y_j}(t_2)$ for some $j \in \{1, \ldots, k\}$. If $\pi^N_{X_i}(t_1) \neq \pi^N_{X_i}(t_2)$, then $\theta_r(\phi(X_i)) =$ *false* and θ_r makes φ' *true*. If $\pi^N_{Y_j}(t_1) = \pi^N_{Y_j}(t_2)$, then $\theta_r(\phi(Y_j)) =$ *true* and θ_r makes φ' *true* again. □

We are now prepared to show the equivalence between *2)* and *3)*. Suppose *2)* does not hold. Then there are $t_1, t_2 \in dom(N)$ such that $t_1 \neq t_2$ and $\models_{\{t_1,t_2\}} \sigma$ for all $\sigma \in \Sigma$, but not $\models_{\{t_1,t_2\}} \varphi$. According to Lemma 21 we know that $\models_{\theta_{\{t_1,t_2\}}} \sigma'$ for all $\sigma' \in \Sigma'$ and not $\models_{\theta_{\{t_1,t_2\}}} \varphi'$. Since t_1 and t_2 are distinct it follows also that $\models_{\theta_{\{t_1,t_2\}}} \phi_N$.

We will show that $\models_{\theta_{\{t_1,t_2\}}} \Sigma_N^{\mathcal{J}}$. Let $\neg\phi(U) \vee \phi(V) \in \Sigma_N^{\mathcal{J}}$. According to the definition of $\Sigma_N^{\mathcal{J}}$ we have $V \leq U$. Suppose that $\theta_{\{t_1,t_2\}}(\phi(U)) = \mathit{true}$. Then $\pi_U^N(t_1) = \pi_U^N(t_2)$ according to the definition of the truth assignment $\theta_{\{t_1,t_2\}}$. Since $V \leq U$ it follows that $\pi_V^N(t_1) = \pi_V^N(t_2)$ holds as well. This means, however, that $\theta_{\{t_1,t_2\}}(\phi(V)) = \mathit{true}$, too.

We will show that $\models_{\theta_{\{t_1,t_2\}}} \Sigma_N^{\mathcal{B}}$. Let $\neg\phi(O) \vee \phi(X) \vee \phi(Y) \in \Sigma_N^{\mathcal{B}}$. Suppose that $\theta_{\{t_1,t_2\}}(\phi(O)) = \mathit{true}$. Then $\pi_O^N(t_1) = \pi_O^N(t_2)$ according to the definition of the truth assignment $\theta_{\{t_1,t_2\}}$. We conclude, by Lemma 10, that $\pi_X^N(t_1) = \pi_X^N(t_2)$ or $\pi_Y^N(t_1) = \pi_Y^N(t_2)$ holds.

Consequently, φ' is not logically implied by $\Sigma' \cup \Sigma_N^{\mathcal{J}} \cup \Sigma_N^{\mathcal{B}} \cup \{\phi_N\}$ as witnessed by $\theta_{\{t_1,t_2\}}$. That means *3)* does not hold and it remains to show that *2)* implies *3)*.

Suppose *3)* does not hold. Then there is some truth assignment θ which makes every formula in $\Sigma' \cup \Sigma_N^{\mathcal{J}} \cup \Sigma_N^{\mathcal{B}} \cup \{\phi_N\}$ *true*, but makes φ' *false*. It is now sufficient to find some $r = \{t_1, t_2\} \subseteq \mathit{dom}(N)$ such that $t_1 \neq t_2$ and $\theta = \theta_r$. In this case, Lemma 21 shows that $\models_r \sigma$ for all $\sigma \in \Sigma$ and not $\models_r \varphi$, i.e., *2)* does not hold.

Let $\mathcal{X} = \{X \in \mathcal{J}(N) \mid \theta(\phi(X)) = \mathit{true}\} \subseteq \mathcal{J}(N)$. The set $\mathcal{X}$ has several properties:

1. $\mathcal{J}(N) - \mathcal{X} \neq \emptyset$: since $\models_\theta \phi_N$ it follows that $\theta(\phi(X)) = \mathit{false}$ for some $\leq$-maximal join-irreducible $X \in \mathcal{J}(N)$.

2. $\mathcal{X}$ is closed downwards with respect to $\leq$ in $\mathcal{J}(N)$: let $X \in \mathcal{X}$ and $Y \leq X$ with $Y \in \mathcal{J}(N)$. Since $X \in \mathcal{X}$ we have $\theta(\phi(X)) = \mathit{true}$. As $\models_\theta \Sigma_N^{\mathcal{J}}$ we also have $\models_\theta \neg\phi(X) \vee \phi(Y)$. Consequently, $\theta(\phi(Y)) = \mathit{true}$ which means that $Y \in \mathcal{X}$ holds as well.

3. $\forall O \in N^{\oplus} \cap \mathcal{X}. \forall X, Y \in \mathcal{J}(N)$ such that X, Y are branched with respect to O we have $X \in \mathcal{X}$ or $Y \in \mathcal{X}$: $O \in \mathcal{X}$ implies that $\theta(\phi(O)) = \mathit{true}$. Since $\models_\theta \neg\phi(O) \vee \phi(X) \vee \phi(Y)$ holds we have $\theta(\phi(X)) = \mathit{true}$ or $\theta(\phi(Y)) = \mathit{true}$. Hence, $X \in \mathcal{X}$ or $Y \in \mathcal{X}$.

According to these properties Lemma 12 shows that there are distinct $t_1, t_2 \in \mathit{dom}(N)$ such that for all $X \in \mathcal{J}(N)$ we have: $\pi_X^N(t_1) = \pi_X^N(t_2)$ if and only if $X \in \mathcal{X}$. Let $r = \{t_1, t_2\}$. We conclude,

$$\begin{aligned} \theta_r(\phi(X)) = \mathit{true} \quad &\Leftrightarrow \quad \pi_X^N(t_1) = \pi_X^N(t_2) && \text{(by Definition of } \theta_r) \\ &\Leftrightarrow \quad X \in \mathcal{X} && \text{(by Definition of } r) \\ &\Leftrightarrow \quad \theta(\phi(X)) = \mathit{true} && \text{(by Definition of } \mathcal{X}) \end{aligned}$$

This completes the proof of Theorem 20.

We illustrate the equivalence between weak functional dependencies in complex value databases and propositional clauses by an example.

Example 20. Consider Σ, φ_1 and φ_2 from Example 19. We will now consider the implication problems whether Σ implies φ_1 and φ_2, respectively, from a logical point of view. First, we define the following mapping ϕ from join-irreducibles to propositional variables as follows:

- $\phi(K(M(A \oplus \lambda_B), \lambda_P)) = V_1$, $\phi(K(M(\lambda_A \oplus B), \lambda_P)) = V_2$,
- $\phi(K(\lambda_M, P[Q(C \oplus \lambda_D)])) = V_3$, $\phi(K(\lambda_M, P[Q(\lambda_C \oplus D)])) = V_4$,
- $\phi(K(M(\lambda_A \oplus \lambda_B), \lambda_P)) = V_5$, $\phi(K(\lambda_M, P[Q(\lambda_C \oplus \lambda_D)])) = V_6$,
- $\phi(K(\lambda_M, P[\lambda_Q])) = V_7$, and $\phi(K(\lambda_M, \lambda_P)) = V_8$.

We then have $\Sigma' = \{\neg V_7 \vee V_5\}, \varphi_1' = \neg V_3 \vee V_1$ and $\varphi_2' = \neg V_3 \vee V_1 \vee V_2$. Moreover, we obtain

$$\Sigma_N^{\mathcal{J}} = \{\neg V_1 \vee V_5, \neg V_2 \vee V_5, \neg V_3 \vee V_6, \neg V_4 \vee V_6, \neg V_5 \vee V_8, \neg V_6 \vee V_7, \neg V_7 \vee V_8\}$$

and

$$\Sigma_N^{\mathcal{B}} = \{\neg V_5 \vee V_1 \vee V_2\}$$

as well as $\phi_N = \neg V_1 \vee \neg V_2 \vee \neg V_3 \vee \neg V_4$.

Recall that φ_1 is not implied by Σ as witnessed by the tuples t_1, t_2 from Example 19. Accordingly, φ_1' is not logically implied by $\Sigma' \cup \Sigma_N^{\mathcal{J}} \cup \Sigma_N^{\mathcal{B}} \cup \{\phi_N\}$. In fact, the truth assignment θ that assigns *false* to V_1 and *true* to all the remaining variables satisfies all formulae in $\Sigma' \cup \Sigma_N^{\mathcal{J}} \cup \Sigma_N^{\mathcal{B}} \cup \{\phi_N\}$ but makes φ_1' *false*. Notice the strong correspondence between $\{t_1, t_2\}$ and θ: t_1 and t_2 agree on precisely those join-irreducibles whose associated variables are evaluated to *true* by θ. However, there is no truth assignment that satisfies all formulae in $\Sigma' \cup \Sigma_N^{\mathcal{J}} \cup \Sigma_N^{\mathcal{B}} \cup \{\phi_N\}$ but violates φ_2'. □

4.3 Encoding with Relational Weak Functional Dependencies

A direct consequence of the Equivalence Theorem 20 is that relational database design solutions can be applied to problems for databases that are not in first normal form (and vice versa). We will now make this correspondence explicit. In particular, the relationship implies that the lower bounds on the time-complexity of the implication problem in relational databases also apply to nested databases.

Let $N \in \mathcal{N}$. We interpret the join-irreducible elements of N as attributes in a relation schema, i.e., the corresponding relation schema of N is $R_N = \mathcal{J}(N)$. Let Σ be a set of wFDs on N. Each wFD φ: $\mathcal{X} \rightarrow_w \mathcal{Y}$ in Σ is mapped to the relational wFD φ': $\max(\mathcal{X}) \rightarrow_w \min(\mathcal{Y})$. Therefore, the corresponding set of relational wFDs is

$$\begin{aligned}\Sigma' = &\{\sigma' \mid \sigma \in \Sigma\} \cup \{U \rightarrow_w V \mid U, V \in \mathcal{J}(N), U \text{ covers } V\} \cup \\ &\{O \rightarrow_w \{X, Y\} \mid X, Y \in \mathcal{J}(N) \text{ are branched with respect to } O \in N^{\oplus}\}.\end{aligned}$$

Corollary 22. *Let $N \in \mathcal{N}$, and $\Sigma \cup \{\varphi\}$ be a set of wFDs on N. Then φ is implied by Σ if and only if φ' is implied by Σ' on R_N.* □

Example 21. Recall Example 20 again. We can view the propositional variables as attributes of the relation schema $R = \{V_1, \ldots, V_8\}$. The corresponding set of weak functional dependencies is

$$\Sigma' = \{V_7 \to V_5\} \cup \{V_1 \to V_5, V_2 \to V_5, V_3 \to V_6, V_4 \to V_6, \\ V_5 \to V_8, V_6 \to V_7, V_7 \to V_8\} \cup \{V_5 \to \{V_1, V_2\}\}.$$

Notice that $\{V_1, V_2, V_3, V_4\}$ forms a minimal key for R with respect to Σ'. The functional dependency $V_3 \to V_1$ is not implied by Σ' (a counterexample relation consists of two tuples that differ on the attribute V_1 and coincide on all other attributes). However, the weak functional dependency $V_3 \to \{V_1, V_2\}$ is implied by Σ'. □

5 Conclusion and Future Work

We have studied weak functional dependencies in relational and complex-value databases with record, list, and disjoint union constructor. While these constraints can be specified in a way similar to traditional functional dependencies they enable a database designer to express every propositional sentence instead of just Horn clauses. Our first major contribution is an axiomatisation of weak functional dependencies in the presence of complex data elements that can be derived by an arbitrary finite number of recursive nestings by record, list and disjoint union constructors. The second contribution establishes an analogy between the implication of weak functional dependencies and propositional formulae in classical propositional logic. Hence, well-studied reasoning tools become applicable to design problems for databases in non-first normal form. We believe that data administrators who are experienced in specifying functional dependencies will find it not much more difficult to specify weak functional dependencies. Notice that this corresponds to the use of clauses instead of arbitrary propositional formulae in propositional logic.

An interesting extension of this research is to incorporate set and multiset constructor into the type system without considering restructuring rules. It is known that join-irreducibles do not suffice in the presence of these constructors in order to obtain a correspondence between the implication of functional dependencies and propositional Horn clauses [Hartmann and Link, 2006; Hartmann and Link, 2008]. However, the interactions between set/multiset and disjoin union constructors are not obvious at all. Furthermore, it would also be interesting to include identifiers and the reference constructor into the treatment. It is also worthwhile to extend this research to other classes of dependencies and data formats, e.g. to multivalued dependencies in partial databases [Lien, 1982;

Link, 2008] and full-hierarchical dependencies in undetermined universes [Hartmann et al., 2007]. Moreover, multivalued dependencies [Fagin, 1977b] have also been studied in the presence of record and list constructor [Hartmann et al., 2006b]. It would be interesting to extend this research to the presence of the disjoint union constructor, too.

Acknowledgement

This research is supported by the Marsden Fund Council from Government funding, administered by the Royal Society of New Zealand.

References

[Abiteboul et al., 1995] Abiteboul, S., Hull, R., and Vianu, V. (1995). *Foundations of Databases*. Addison-Wesley.

[Anderson, 1987] Anderson, I. (1987). *Combinatorics of finite sets*. Oxford Science Publications. The Clarendon Press Oxford University Press, New York.

[Arenas and Libkin, 2004] Arenas, M. and Libkin, L. (2004). A normal form for XML documents. *ACM Trans. Database Syst.*, 29(1):195–232.

[Bray et al., 2006] Bray, T., Paoli, J., Sperberg-McQueen, C. M., Maler, E., and Yergeau, F. (2006). Extensible markup language (XML) 1.0 (Fourth Edition) W3C Recommendation, Aug. 2006. http://www.w3.org/TR/xml/.

[Cook, 1971] Cook, S. (1971). The complexity of theorem-proving procedures. In *STOC*, pages 151–158.

[Demetrovics and Gyepesi, 1981] Demetrovics, J. and Gyepesi, G. (1981). On functional dependency and some generalisations of it. *Acta Cybern.*, 5:295–305.

[Dowling and Gallier, 1984] Dowling, W. and Gallier, J. (1984). Linear-time algorithms for testing the satisfiability of propositional horn formulae. *J. Log. Program.*, 1(3):267–284.

[Enderton, 2001] Enderton, H. (2001). *A mathematical introduction to logic: Second Edition*. Academic Press.

[Fagin, 1977a] Fagin, R. (1977a). Functional dependencies in a relational data base and propositional logic. *IBM Journal of Research and Development*, 21(6):543–544.

[Fagin, 1977b] Fagin, R. (1977b). Multivalued dependencies and a new normal form for relational databases. *ACM Trans. Database Syst.*, 2(3):262–278.

[Fagin and Vardi, 1986] Fagin, R. and Vardi, M. Y. (1986). The theory of data dependencies: a survey. In *Mathematics of Information Processing: Proceedings of Symposia in Applied Mathematics*, pages 19–71. American Mathematical Society.

[Graetzer, 1998] Graetzer, G. (1998). *General Lattice Theory*. Birkhauser.

[Hara and Davidson, 1999] Hara, C. S. and Davidson, S. B. (1999). Reasoning about nested functional dependencies. In *PODS*, pages 91–100. ACM.

[Hartmann et al., 2007] Hartmann, S., Köhler, H., and Link, S. (2007). Full hierarchical dependencies in fixed and undetermined universes. *Ann. Math. Artif. Intell.*, 50(1-2):195–226.

[Hartmann and Link, 2006] Hartmann, S. and Link, S. (2006). Deciding implication for functional dependencies in complex-value databases. *Theor. Comput. Sci.*, 364(2):212–240.

[Hartmann and Link, 2008] Hartmann, S. and Link, S. (2008). Characterising nested database dependencies by fragments of propositional logic. *Ann. Pure Appl. Logic*, 152(1-3):84–106.

[Hartmann et al., 2004] Hartmann, S., Link, S., and Schewe, K.-D. (2004). Weak functional dependencies in higher-order datamodels. In *3rd International Symposium on Foundations of Information and Knowledge Systems (FoIKS)*, number 2942 in Lecture Notes in Computer Science, pages 116–133. Springer.

[Hartmann et al., 2005] Hartmann, S., Link, S., and Schewe, K.-D. (2005). Functional dependencies over XML documents with DTDs. *Acta Cybern.*, 17(1):153–171.

[Hartmann et al., 2006a] Hartmann, S., Link, S., and Schewe, K.-D. (2006a). Axiomatisations of functional dependencies in the presence of records, lists, sets and multisets. *Theor. Comput. Sci.*, 355(2):167–196.

[Hartmann et al., 2006b] Hartmann, S., Link, S., and Schewe, K.-D. (2006b). Functional and multivalued dependencies in nested databases generated by record and list constructor. *Ann. Math. Artif. Intell.*, 46(1-2):114–164.

[Hirsch and Kojevnikov, 2005] Hirsch, E. and Kojevnikov, A. (2005). UnitWalk: A new SAT solver that uses local search guided by unit clause elimination. *Ann. Math. Artif. Intell.*, 43(1):91–111.

[Levene, 1992] Levene, M. (1992). *The Nested Universal Relation Database Model*. Springer.

[Levine, 1973] Levine, L. (1973). Universal search problems. *Problemy Peredachi Informatsii*, 9(3):265–266.

[Lien, 1982] Lien, Y. E. (1982). On the equivalence of data models. *J. ACM*, 29(2):333–363.

[Link, 2008] Link, S. (2008). On the implication of multivalued dependencies in partial database relations. *Int. J. Found. Comput. Sci.*, 19(3):691–715.

[McKinsey and Tarski, 1946] McKinsey, J. C. C. and Tarski, A. (1946). On closed elements in closure algebras. *Annals of Mathematics*, 47:122–146.

[Prasad et al., 2005] Prasad, M., Biere, A., and Gupta, A. (2005). A survey of recent advances in SAT-based formal verification. *STTT*, 7(2):156–173.

[Sagiv et al., 1981] Sagiv, Y., Delobel, C., Parker Jr., D. S., and Fagin, R. (1981). An equivalence between relational database dependencies and a fragment of propositional logic. *J. ACM*, 28(3):435–453.

[Sagiv et al., 1987] Sagiv, Y., Delobel, C., Parker Jr., D. S., and Fagin, R. (1987). Correction to "An equivalence between relational database dependencies and a fragment of propositional logic". *J. ACM*, 34(4):1016–1018.

[Sali and Schewe, 2006] Sali, A. and Schewe, K.-D. (2006). Counter-free keys and functional dependencies in higher-order datamodels. *Fundam. Inform.*, 70(3):277–301.

[Schewe, 2005] Schewe, K.-D. (2005). Functional dependencies with counting on trees. *J. UCS*, 11(12):2063–2075.

[Thalheim, 1991] Thalheim, B. (1991). *Dependencies in Relational Databases.* Teubner-Verlag.

[Vincent et al., 2004] Vincent, M., Liu, J., and Liu, C. (2004). Strong functional dependencies and their application to normal forms in xml. *ACM Trans. Database Syst.*, 29(3):445–462.

[Weddell, 1992] Weddell, G. E. (1992). Reasoning about functional dependencies generalized for semantic data models. *ACM Trans. Database Syst.*, 17(1):32–64.