The Nested List Normal Form for Functional and Multivalued Dependencies

Sven Hartmann and Sebastian Link*

Information Science Research Centre, Dept of Information Systems,
Massey University, Palmerston North, New Zealand
{s.hartmann, s.link}@massey.ac.nz

Abstract. The Nested List Normal Form is proposed as a syntactic normal form for semantically well-designed database schemata obtained from any arbitrary finite nesting of records and lists. The Nested List Normal Form is defined in terms of functional and multivalued dependencies, independent from any specific data model, and generalises the well-known Fourth Normal Form from relational databases in order to capture more application domains.

1 Introduction

An important issue associated with the use of any databases is the correct structure or design of data to be used. Several criteria, referred to as *normal forms*, have been proposed as conditions for database schemata that a database design should satisfy to ensure an absence of processing difficulties with the database. These normal forms give a database designer unambiguous guidelines in deciding which databases are good in the quest to avoid bad designs that have redundancy problems and update anomalies. Such normal forms have already been introduced in [12] by Codd himself. In general, they are dependent on the type of integrity constraints or rules which apply to data items within the database. Important classes of integrity constraints are *functional dependencies* (FDs) [12] and *Multivalued dependencies* (MVDs) [14]. FDs and MVDs cause difficulties such as redundancy in the representation of data and update anomalies. The Boyce-Codd normal form (BCNF) was proposed to overcome these difficulties with FDs [13], and Fagin introduced the Fourth Normal form (4NF) to deal with the more general class of FDs and MVDs [14]. Later on, after the notions of redundancy and update anomaly had been clarified and formalised, it was shown that BCNF (4NF) precisely captures those relation schemata that are free from redundancies and update anomalies in terms of FDs (FDs and MVDs) [7, 15, 35]. Normalisation has been studied in the context of other data models as well. There are several normal form proposals for the nested relational data model, and a detailed comparison can be found in [30]. Recently, the issue of normalisation has been revived in the context of XML [2, 3, 36, 37]. XNF is defined in terms of FDs that are based on a path-like notion in DTDs and do not enjoy a finite

* Sebastian Link was supported by Marsden Funding, Royal Society of New Zealand.

J. Dix and S.J. Hegner (Eds.): FoIKS 2006, LNCS 3861, pp. 137–158, 2006.

ground axiomatisation [2]. In [3] techniques from information theory are used to provide justifications for several normal forms and normalisation algorithms. MVDs have also been introduced into the context of XML and an extension of 4NF has been proposed [37]. Apart from [18, 31], who consider set equality in the nested relational data model, all previous approaches to defining constraints in advanced data formats do not consider equality on complex objects such as lists, sets or multisets, and are therefore unable to express important semantic information that occurs in many applications.

Several researchers have remarked that classical database design problems need to be revisited in new data formats [3, 32, 34, 35]. Biskup [8, 9] has listed two particular challenges for database design theory: finding a unifying framework and extending achievements to deal with advanced database features such as complex object types. We propose to classify data models according to the type constructors they support. Thus, the relational data model can be captured by a single application of the record type, arbitrary nesting of record and set constructor covers aggregation and grouping which are fundamental to many semantic data models as well as the nested relational data model [1, 24]. The Entity-Relationship Model and its extensions require record, set and (disjoint) union constructor [11, 33]. A minimal set of type constructors supported by any object-oriented data model includes records, lists, sets and multisets (bags) [5]. Genomic sequence data models call for support of records, lists and sets [28]. Finally, XML requires at least record (concatenation), list (Kleene Closure), union (optionality), and reference constructor [10].

In this paper we study database design in the presence of record and list constructor with respect to functional and multivalued dependencies. It is our goal to achieve an adequate extension of 4NF from relational databases, and to actually demonstrate what this extension achieves in terms of characterising the absence of adequate extensions of the notions of redundancies and update anomalies. Our studies will be based on an abstract data model that defines a database schema as an arbitrarily nested attribute where nesting applies record and list constructor. It is our intention not to focus on any specific data model in order to place emphasis on the type constructors themselves. Dependencies are defined in terms of subschemata of the underlying database schema. This approach provides a mathematically well-founded framework that is sufficiently flexible and powerful to study design problems for different classes of constraints. The fact that the set of all subschemata of some fixed database schema carries the structure of a Brouwerian algebra turns out to precisely accommodate the needs of multivalued dependencies.

Throughout the article we will apply the theory to an example from image processing that we introduce now. Digital halftoning plays a key role in almost every discipline that involves printing and displaying. All newspapers, magazines, and books are printed with digital halftoning. One method to perform digital halftoning is error diffusion [16, 25, 26]: once a pixel has been quantised, thus introducing some error, this error should affect the quantisation of the pixels in the region of its neighbours. Digital halftoning is an application of the matrix rounding problem [4]. The problem is to convert a continuous-tone image into a binary one that looks similar. The input matrix A represents a digital (gray) image, where a_{ij} represents the brightness level of the (i,j)-pixel in the $n \times n$ pixel grid. Typically, n is between 256 and 4096, and a_{ij} is an integral multiple of $\frac{1}{256}$: this means that we use 256 brightness levels. If we want to send an image using fax or print it out by a dot or ink-jet printer, brightness levels available are limited. Instead, we replace the input matrix A by an integral matrix B so that each pixel uses only two brightness levels. Here, it is important that B looks similar to A; in other words, B should be an approximation of A. In this sense, an approximation of input matrix A is a {0,1}-matrix B that minimises the distance $\left| \sum_{(i,j)\in R} a_{i,j} - \sum_{(i,j)\in R} b_{i,j} \right|$ for all $R \in \mathcal{R}$. In this formulae $\mathcal{R}$ denotes the set of regions of neighbors, for instance the set of all pairs of indices that denote $2 \times 1, 1 \times 2$ and 2×2 submatrices. A region has therefore one of the following forms

$$\begin{array}{|c|c|}\hline a & b \\ \hline c & d \\ \hline \end{array} \qquad \begin{array}{|c|c|}\hline a & b \\ \hline \end{array} \qquad \begin{array}{|c|}\hline a \\ \hline b \\ \hline \end{array} \quad ,$$

and can be represented as a list of either two or four elements. The regions may have all different kinds of shapes in practice. In order to make the example more illustrative, we assume from now on that the input matrix has entries in $\{0, \frac{1}{2}, 1\}$, i.e., uses three brightness levels. Input regions can be best approximated by a number of different output regions. All inputs with overall brightness $\frac{1}{2}$ and length two, i.e. $[0, \frac{1}{2}]$ or $[\frac{1}{2}, 0]$, could be mapped to any of [0,1], [1,0] or [0,0], each of which has distance $\frac{1}{2}$. In this sense, the set of input sequences $(\{[0, \frac{1}{2}], [\frac{1}{2}, 0]\})$ is determined by the overall brightness of the input sequences $(\frac{1}{2})$ and the length of the input sequence (2), independently of the set of output sequences $(\{[0,1],[1,0],[0,0]\})$. This is true for any inputs and outputs, e.g., all inputs with overall brightness $\frac{3}{2}$ and length four such as $[0, 0, 1, \frac{1}{2}]$ can be mapped to any of [0,0,0,1], [0,0,1,0], [0,1,0,0], [1,0,0,0], [0,0,1,1], [0,1,0,1], [1,0,0,1], [0,1,1,0], [1,0,1,0], [1,1,0,0].

Consider a database which stores input and output sequences together with the overall brightness of the input sequence. It is then desirable to find a {0,1}-matrix B that has for every of the possible regions of input matrix A a corresponding output region that are stored together as an entry in the database. The input matrix $A = \begin{pmatrix} 0 & 0 \\ \frac{1}{2} & \frac{1}{2} \end{pmatrix}$ has for instance the approximation $B = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$. Every 2×2 matrix has five input sequences and the mappings that produce B from A are as follows: $[0,0] \mapsto [0,0]$, $[\frac{1}{2}, \frac{1}{2}] \mapsto [0,1]$, $[0, \frac{1}{2}] \mapsto [0,0]$ (left column), $[0, \frac{1}{2}] \mapsto [0,1]$ (right column) and $[0, 0, \frac{1}{2}, \frac{1}{2}] \mapsto [0,0,0,1]$.

The matrix $\begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$, however, is not an approximation of A as the sequence $[\frac{1}{2}, \frac{1}{2}]$ should not be mapped to $[0,0]$.

Constraints that a database designer may choose to specify for this application are the following:

1. The length of the input sequence determines the length of the output sequence, and vice versa.
2. The overall brightness and length of the input sequence together determine the set of all input sequences independently from the set of the output sequences.

The example illustrates a typical scenario where list equality occurs in a constraint specification. We will formalise all parts of this example during the course of the paper, and see whether the suggested design is appropriate with respect to the constraints specified.

2 A Summary of Previous Work

2.1 The Complex-Value Data Model

This section introduces a data model based on the nesting of attributes and subtyping. It may be used to provide a framework for the study of type constructors such as records, lists, sets, multisets, unions and references. This article, however, focuses on records and lists. In terms of XML the reader may notice that we deal with a slightly extended fragment of DTDs in which only concatenation and Kleene closure are allowed. However, the expressiveness of our constraints is different from previous approaches as we are particularly interested in list equality.

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 set $\mathcal{L}$ of labels, and assume that the symbol λ is neither a flat attribute nor a label, i.e., $\lambda \notin \mathcal{U} \cup \mathcal{L}$. Moreover, flat attributes are not labels and vice versa, i.e., $\mathcal{U} \cap \mathcal{L} = \emptyset$.

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 $\mathcal{N}\mathrm{A}(\mathcal{U}, \mathcal{L})$ of *nested attributes over* $\mathcal{U}$ *and* $\mathcal{L}$ is the smallest set satisfying the following conditions: $\lambda \in \mathcal{N}\mathrm{A}(\mathcal{U}, \mathcal{L})$, $\mathcal{U} \subseteq \mathcal{N}\mathrm{A}(\mathcal{U}, \mathcal{L})$, for $L \in \mathcal{L}$ and $N_1, \ldots, N_k \in \mathcal{N}\mathrm{A}(\mathcal{U}, \mathcal{L})$ with $k \geq 1$ we have $L(N_1, \ldots, N_k) \in \mathcal{N}\mathrm{A}(\mathcal{U}, \mathcal{L})$, for $L \in \mathcal{L}$ and $N \in \mathcal{N}\mathrm{A}(\mathcal{U}, \mathcal{L})$ we have $L[N] \in \mathcal{N}\mathrm{A}(\mathcal{U}, \mathcal{L})$. We call λ *null attribute*, $L(N_1, \ldots, N_k)$ *record-valued attribute* and $L[N]$ *list-valued attribute*. We will use upper-case letters from the middle of the alphabet such as $N, M, etc.$ to refer to nested attributes. From now on, we assume that a set $\mathcal{U}$ of attribute names, and a set $\mathcal{L}$ of labels is fixed, and write $\mathcal{N}\mathrm{A}$ instead of $\mathcal{N}\mathrm{A}(\mathcal{U}, \mathcal{L})$. We may use the nested attribute

$$\textsc{Halftoning}(\text{Brightness},\textsc{Input}[\text{Level}],\textsc{Output}[\text{Bit}])$$

as a database schema for instances of the digital halftoning database described in the introduction. Labels are HALFTONING, INPUT and OUTPUT, and flat attribute names are Brightness, Level and Bit. The domain of the flat attribute Level is $\{0, \frac{1}{2}, 1\}$ and the domain of the flat attribute Bit is $\{0, 1\}$.

In general, 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 $N \in \mathcal{N}\mathrm{A}$. For a nested attribute $N \in \mathcal{N}\mathrm{A}$ we define the *domain* $dom(N)$ as follows: $dom(\lambda) = \{ok\}$, $dom(L(N_1, \ldots, N_k)) = \{(v_1, \ldots, v_k) \mid v_i \in dom(N_i)$ for $i = 1, \ldots, k\}$, i.e., the set of all k-tuples $(v_1, \ldots, v_k)$ with $v_i \in dom(N_i)$ for all $i = 1, \ldots, k$, and $dom(L[N]) = \{[v_1, \ldots, v_n] \mid v_i \in dom(N)$ for $i = 1, \ldots, n\}$, i.e., the set of all finite lists with elements in $dom(N)$. The empty list is denoted by $[\,]$. For instance, the domain of $\textsc{Input}[\lambda]$ is the set of all finite lists consisting of elements *ok*, i.e., $\{[\,], [ok], [ok, ok], \ldots\}$. The nested attribute $\textsc{Input}[\lambda]$ therefore still tells us how long the lists over $\textsc{Input}[\text{Level}]$ are. The value *ok* can be interpreted as the null value “some information exists, but is currently omitted”.

The replacement of attributes by the null attribute λ decreases the amount of information modelled. This fact allows one to introduce an order between nested attributes. The *subattribute relation* $\leq$ on the set of nested attributes $\mathcal{N}\mathrm{A}$ over $\mathcal{U}$ and $\mathcal{L}$ is defined by the following rules, and the following rules only: $N \leq N$, $\lambda \leq A$ for all flat attributes $A \in \mathcal{U}$, $\lambda \leq N$ for all list-valued attributes N, $L(N_1, \ldots, N_k) \leq L(M_1, \ldots, M_k)$ whenever $N_i \leq M_i$ for all $i = 1, \ldots, k$, and $L[N] \leq L[M]$ whenever $N \leq M$. 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 1 ([22]). *The subattribute relation is a partial order on nested attributes.* □

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 $\pi^N_M : dom(N) \to dom(M)$ in case $M \leq N$ holds. For $M \leq N$ the *projection function* $\pi^N_M : dom(N) \to dom(M)$ is defined as follows:

- if $N = M$, then $\pi^N_M = id_{dom(N)}$ is the identity on $dom(N)$,
- if $M = \lambda$, then $\pi^N_\lambda : dom(N) \to \{ok\}$ is the constant function that maps every $v \in dom(N)$ to *ok*,
- if $N = L(N_1, \ldots, N_k)$ and $M = L(M_1, \ldots, M_k)$, then $\pi^N_M = \pi^{N_1}_{M_1} \times \cdots \times \pi^{N_k}_{M_k}$ which maps every tuple $(v_1, \ldots, v_k) \in dom(N)$ to $(\pi^{N_1}_{M_1}(v_1), \ldots, \pi^{N_k}_{M_k}(v_k)) \in dom(M)$, and
- if $N = L[N']$ and $M = L[M']$, then $\pi^N_M : dom(N) \to dom(M)$ maps every list $[v_1, \ldots, v_n] \in dom(N)$ to the list $[\pi^{N'}_{M'}(v_1), \ldots, \pi^{N'}_{M'}(v_n)] \in dom(M)$.

The set $Sub(N)$ of *subattributes* of N is $Sub(N) = \{M \mid M \leq N\}$. Note that $Sub(N)$ is always finite. Lemma 1 shows that the restriction of $\leq$ to $Sub(N)$ is a partial order on $Sub(N)$. We study the algebraic structure of $Sub(N)$. A *Brouwerian algebra* [29] 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 [29]. The *join* $X \sqcup_N Y$, *meet* $X \sqcap_N Y$ and *pseudo-difference* $X \dot{-}_N Y$ of X and Y in $Sub(N)$ are completely determined by the subattribute order $\leq$. We use $Y_N^{\mathcal{C}} = N \dot{-} Y$ to denote the *Brouwerian complement* of Y in $Sub(N)$. The following theorem generalises the fact that $(\mathcal{P}(R), \subseteq, \cup, \cap, -, \emptyset, R)$ is a Boolean algebra for a relation schema R.

Theorem 1 ([22]). *$(Sub(N), \leq, \sqcup_N, \sqcap_N, \dot{-}_N, N)$ forms a Brouwerian algebra for every $N \in \mathcal{N}A$.* □

In order to simplify notation, occurrences of λ in a record-valued attribute are usually omitted if this does not cause any ambiguities. That is, the subattribute $L(M_1, \ldots, M_k) \leq L(N_1, \ldots, N_k)$ is abbreviated by $L(M_{i_1}, \ldots, M_{i_l})$ where $\{M_{i_1}, \ldots, M_{i_l}\} = \{M_j : M_j \neq \lambda_{N_j}$ and $1 \leq j \leq k\}$ and $i_1 < \cdots < i_l$. If $M_j = \lambda_{N_j}$ for all $j = 1, \ldots, k$, then we use λ instead of $L(M_1, \ldots, M_k)$. The subattribute Halftoning(λ,Input[λ],λ) is abbreviated by Halftoning(Input[λ]). However, the subattribute $L(A, \lambda)$ of $L(A, A)$ cannot be abbreviated by $L(A)$ since this may also refer to $L(\lambda, A)$. If the context allows, we omit the index N from the operations $\sqcup_N, \sqcap_N, \dot{-}_N, (\cdot)_N^{\mathcal{C}}$ and from λ_N. The Brouwerian algebra for Halftoning(Brightness,Input[Level],Output[Bit]) is illustrated in Figure 1.

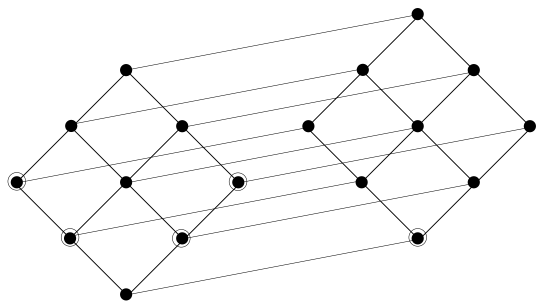

Fig. 1. Brouwerian algebra of Halftoning(Brightness,Input[Level],Output[Bit])

Fundamental to lists is the following fact: if $\pi_X^N(t_1) = \pi_X^N(t_2)$ and $\pi_Y^N(t_1) = \pi_Y^N(t_2)$, then also $\pi_{X \sqcup Y}^N(t_1) = \pi_{X \sqcup Y}^N(t_2)$ for any $t_1, t_2 \in dom(N)$ [22]. This suggests to focus on join-irreducible elements of $(Sub(N), \leq, \sqcup, \sqcap, \lambda_N)$. 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 $a = c$. Let $\mathcal{B}(N)$ denote the set of join-irreducible elements of $(Sub(N), \leq, \sqcup, \sqcap, \dot{-}, N)$, and $\mathcal{B}_{\mathcal{M}}(N)$ the maximal elements of $\mathcal{B}(N)$ with respect to $\leq$. The join-irreducibles of Halftoning(Brightness,Input[Level],Output[Bit]) are circled in Figure 1.

2.2 An Axiomatisation for FDs and MVDs

In this section we repeat previous definitions and results [20, 22].The data model allows us to introduce a natural extension of the notion of FDs and MVDs from the relational data model.

A *functional dependency (FD)* on the nested attribute N is an expression of the form $X \to Y$ where $X, Y \in Sub(N)$. A set $r \subseteq dom(N)$ *satisfies* the functional dependency $X \rightarrow Y$ on N, denoted by $\models_r X \rightarrow Y$, if and only if $\pi_Y^N(t_1) = \pi_Y^N(t_2)$ whenever $\pi_X^N(t_1) = \pi_X^N(t_2)$ for any $t_1, t_2 \in r$ holds. A *multivalued dependency (MVD)* on N is an expression of the form $X \twoheadrightarrow Y$ where $X, Y \in Sub(N)$. A set $r \subseteq dom(N)$ *satisfies* the multivalued dependency $X \twoheadrightarrow Y$ on N if and only if for all values $t_1, t_2 \in r$ with $\pi_X^N(t_1) = \pi_X^N(t_2)$ there is a value $t \in r$ with $\pi_{X \sqcup Y}^N(t) = \pi_{X \sqcup Y}^N(t_1)$ and $\pi_{X \sqcup Y^{\mathcal{C}}}^N(t) = \pi_{X \sqcup Y^{\mathcal{C}}}^N(t_2)$.

The constraints on Halftoning(Brightness,Input[Level],Output[Bit]), informally described in the introduction, can now be formalised (using abbreviations) as:

$$\text{Halftoning}(\text{Input}[\lambda]) \rightarrow \text{Halftoning}(\text{Output}[\lambda]),$$
$$\text{Halftoning}(\text{Output}[\lambda]) \rightarrow \text{Halftoning}(\text{Input}[\lambda]), \text{ and}$$
$$\text{Halftoning}(\text{Brightness},\text{Input}[\lambda]) \twoheadrightarrow \text{Halftoning}(\text{Input}[\text{Level}]).$$

Fagin proves [14] that relational MVDs "provide a necessary and sufficient condition for a relation to be decomposable into two of its projections without loss of information (in the sense that the original relation is guaranteed to be the join of the two projections)." Let $N \in \mathcal{N}\text{A}$ and $X, Y \in Sub(N)$. Let $r_1 \subseteq dom(X)$ and $r_2 \subseteq dom(Y)$. Then $r_1 \bowtie r_2 = \{t \in dom(X \sqcup Y) \mid \text{ there are } t_1 \in r_1, t_2 \in r_2 \text{ with } \pi_X^{X \sqcup Y}(t) = t_1 \text{ and } \pi_Y^{X \sqcup Y}(t) = t_2\}$ is called the *generalised join* $r_1 \bowtie r_2$ *of* r_1 and r_2. The *projection* $\pi_X(r)$ of $r \subseteq dom(N)$ on $X \in Sub(N)$ is defined as $\{\pi_X^N(t) \mid t \in r\}$.

Theorem 2 ([22]). *Let $N \in \mathcal{N}A$, and $r \subseteq dom(N)$. Then is $X \twoheadrightarrow Y$ satisfied by r if and only if $r = \pi_{X \sqcup Y}(r) \bowtie \pi_{X \sqcup Y^{\mathcal{C}}}(r)$. If r satisfies the FD $X \rightarrow Y$, then $r = \pi_{X \sqcup Y}(r) \bowtie \pi_{X \sqcup Y^{\mathcal{C}}}(r)$.* □

The notions of *implication* ($\models$) and *derivability* ($\vdash_{\mathfrak{R}}$) with respect to a set $\mathfrak{R}$ of inference rules for a class $\mathcal{C}$ of dependencies can be defined analogously to the notions in relational databases [1–pp. 164-168]. Note that finite and unrestricted implication coincide for functional and multivalued dependencies, even in the presence of lists [22]. The notions of *soundness* and *completeness* for a set $\mathfrak{R}$ of inference rules carry over as well. A dependency σ on some nested attribute N is called *trivial* if and only if $\models_r \sigma$ for every $r \subseteq dom(N)$. An FD $X \rightarrow Y$ on N is trivial iff $Y \leq X$ holds, and an MVD $X \twoheadrightarrow Y$ on N is trivial iff $Y \leq X$ or $X \sqcup Y = N$ holds. Note that $X \sqcup Y = N$ iff $Y^{\mathcal{C}} \leq X$. A complete set of inference rules is said to be *minimal* if and only if none of its rules can be omitted without losing completeness.

Theorem 3 ([20, 22]). *The following inference rules*

$\dfrac{}{X \rightarrow Y}\ Y \leq X$	$\dfrac{X \rightarrow Y}{X \rightarrow X \sqcup Y}$	$\dfrac{X \rightarrow Y, Y \rightarrow Z}{X \rightarrow Z}$
(reflexivity axiom)	*(extension rule)*	*(transitivity rule)*
$\dfrac{X \rightarrow Y}{X \twoheadrightarrow Y}$	$\dfrac{X \twoheadrightarrow Y, Y \rightarrow Z}{X \rightarrow (Z \dot{-} Y)}$	$\dfrac{X \twoheadrightarrow Y}{X \rightarrow Y \sqcap Y^{\mathcal{C}}}$
(implication rule)	*(mixed pseudo-transitivity rule)*	*(mixed meet rule)*
$\dfrac{X \twoheadrightarrow Y, Y \twoheadrightarrow Z}{X \twoheadrightarrow (Z \dot{-} Y)}$	$\dfrac{X \twoheadrightarrow Y}{X \twoheadrightarrow Y^{\mathcal{C}}}$	$\dfrac{X \twoheadrightarrow Y, X \twoheadrightarrow Z}{X \twoheadrightarrow (Y \sqcup Z)}$
(pseudo-transitivity rule)	*(Brouwerian complement rule)*	*(multivalued join rule)*

are minimal, sound and complete for the implication of FDs and MVDs in the presence of records and lists. □

Keys play a central role in the retrieval of information because they provide a method by which an element of a database may be identified.

Definition 1. Let $N \in \mathcal{N}A$ be a nested attribute and Σ a set of FDs and MVDs on N. A subattribute $X \in Sub(N)$ is called a *superkey for* N with respect to Σ if and only if $\Sigma \models X \rightarrow N$ holds. In case there is not any proper subattribute $X' < X$ which is also a superkey for N with respect to Σ, we call X a *minimal key for* N with respect to Σ. □

$X \in Sub(N)$ is a superkey if and only if $X \rightarrow N \in \Sigma^+$ by Theorem 3. If $\models_r \Sigma$ for some $r \subseteq dom(N)$ and X is a superkey for N, then $t_1 = t_2$ whenever $\pi_X^N(t_1) = \pi_X^N(t_2)$ for any $t_1, t_2 \in r$. Furthermore, X is a superkey for N if and only if $X^+ = N$ where $X^+ = \bigsqcup\{Z \mid X \rightarrow Z \in \Sigma^+\}$. An FD $X \rightarrow N \in \Sigma^*$ is called a *key dependency* on N with respect to Σ if and only if X is a minimal key for N with respect to Σ. The set of all key dependencies is denoted by Σ_{key}.

3 The Nested List Normal Form

In this section we will investigate normalisation issues in the presence of records and lists in terms of FDs and MVDs. This will extend previous work in which only FDs have been considered [21].

3.1 Three Notions of Redundancy

In relational databases the definition of redundancy is based on viewing FDs and MVDs not only as integrity constraints on a relation, but also as representing the fundamental units of information for retrieving and updating the data in a relation. This interpretation of the semantics of the information stored in a relation was implicit in the original study of normalisation by Codd [12], and has since been used in many aspects of database theory. A relation schema is defined to be redundant with respect to a given set of FDs and MVDs if there exists a relation over the schema which satisfies all these FDs and MVDs and which has at least two tuples which are identical on a fact. If we formalise this notion of redundancy [6] in the framework of nested attributes, then we obtain the following definition. A nested attribute N is *redundant with respect to* a set Σ of FDs and MVDs on N if and only if there is some $r \subseteq dom(N)$ with $\models_r \Sigma$ and there are some $t_1, t_2 \in r$ with $t_1 \neq t_2$ and $\pi^N_{X \sqcup Y}(t_1) = \pi^N_{X \sqcup Y}(t_2)$ for some $X \rightarrow Y \in \Sigma$ or some $X \twoheadrightarrow Y \in \Sigma$ which is not trivial. Intuitively, this notion of redundancy seems to make perfect sense. Take a look at the FD Halftoning(Input[λ]) $\rightarrow$ Halftoning(Output[λ]). This is a non-trivial FD. The elements $(\frac{1}{2}, [0, \frac{1}{2}], [0, 0])$ and $(1, [0, 1], [0, 1])$ coincide on Halftoning(Input[λ],Output[λ]), i.e., the FD causes some redundancy according to the definition above. This example shows that our current definition of redundancy is not really appropriate anymore. That is, the FD Halftoning(Input[λ]) $\rightarrow$

HALFTONING(OUTPUT[Bit]) is not satisfied by the two elements above and, consequently, redundancy would need to be defined in terms of the non-maximal join-irreducible HALFTONING(OUTPUT[λ]). This, however, appears to be impossible as the information in HALFTONING(OUTPUT[λ]) will always be contained in HALFTONING(OUTPUT[Bit]). The point here is that the information in a non-maximal join-irreducible Y cannot be separated from the information in any maximal join-irreducible Z with $Y \leq Z$. We will see further evidence for this in Section 4. This motivates the following definition.

Definition 2. Let $N \in \mathcal{N}A$ be a nested attribute and Σ a set of FDs and MVDs on N. Let $\Sigma_{\text{inev}} \subseteq \Sigma^+$ denote the union of all $X \to Y \in \Sigma^+$ where $Y \leq X$ or $Y \in \mathcal{B}(N) - \mathcal{B}_{\mathcal{M}}(N)$ holds, and all $X \twoheadrightarrow Y \in \Sigma^+$ where $Y \leq X$ or $Y^{\mathcal{C}} \leq X$ or $Y \in \mathcal{B}(N) - \mathcal{B}_{\mathcal{M}}(N)$ holds. The FDs (MVDs) of the closure Σ^+_{inev} of Σ_{inev} under inference with respect to the inference rules from Theorem 3 are called *inevitable FDs (MVDs)* on N with respect to Σ. □

The following lemma characterises inevitable dependencies which are derivable from a given set of FDs and MVDs.

Lemma 2. *Let $N \in \mathcal{N}A$, Σ a set of FDs and MVDs on N. If $X \to Y \in \Sigma^+$, then $X \to Y \in \Sigma^+_{inev}$ if and only if $Y^{\mathcal{CC}} \leq X$. If $X \twoheadrightarrow Y \in \Sigma^+$, then $X \twoheadrightarrow Y \in \Sigma^+_{inev}$ if and only if $Y^{\mathcal{CC}} \leq X$ or $Y^{\mathcal{C}} \leq X$.*

Proof (Sketch). In order to show that $X \to Y \in \Sigma^+_{\text{inev}}$ implies $Y^{\mathcal{CC}} \leq X$, and that $X \twoheadrightarrow Y \in \Sigma^+_{\text{inev}}$ implies $Y^{\mathcal{CC}} \leq X$ or $Y^{\mathcal{C}} \leq X$, one can proceed by induction on the inference length using the inference rules from Theorem 3. The remaining direction is a matter of applying some of these inference rules as well. □

Thus, trivial FDs and MVDs are always inevitable on N with respect to any Σ, but not vice versa. We are now prepared to define a better notion of redundancy for nested attributes *in terms of FDs and MVDs*.

Definition 3. Let Σ be a set of FDs and MVDs on the nested attribute N. We call N *type-1 redundant with respect to* Σ if and only if there is some $r \subseteq dom(N)$ with $\models_r \Sigma$ and there are some distinct $t_1, t_2 \in r$ with $\pi^N_{X \sqcup Y}(t_1) = \pi^N_{X \sqcup Y}(t_2)$ for some FD $X \to Y \in \Sigma$ that is not inevitable on N with respect to Σ or some MVD $X \twoheadrightarrow Y \in \Sigma$ that is not inevitable on N with respect to Σ . □

Definition 3 allows the set of facts to be the subattributes in all the FDs and MVDs which are not inevitable in a user-supplied set of dependencies Σ. However, one may also recognise the symmetrical nature of MVDs and so allow the subattributes in any MVD that can be derived from any MVD in Σ and successive applications of the Brouwerian-complement rule to also be a fact. Finally, the last possibility is to include inferred dependencies and allow subattributes in any FD or MVD that is not inevitable and implied by Σ to be a fact. Intuitively, one would expect that the notion of redundancy is independent of which of these facts is chosen but in general this is not the case, and the proof is by no means immediate. Let Σ' denote the smallest set with the following properties: $\Sigma \subseteq \Sigma'$ and $X \twoheadrightarrow Y^{\mathcal{C}} \in \Sigma'$ whenever $X \twoheadrightarrow Y \in \Sigma'$.

Definition 4. Let N be a nested attribute and Σ a set of FDs and MVDs on N. We call N *type-2(3) redundant with respect to* Σ if and only if there is some $r \subseteq dom(N)$ with $\models_r \Sigma$ and there are some distinct $t_1, t_2 \in r$ with $\pi^N_{X \sqcup Y}(t_1) = \pi^N_{X \sqcup Y}(t_2)$ for some FD $X \to Y \in \Sigma'(\Sigma^+)$ which is not inevitable on N with respect to Σ or some MVD $X \twoheadrightarrow Y \in \Sigma'(\Sigma^+)$ which is not inevitable on N with respect to Σ. □

Let $N = L(A, B, C)$ and $\Sigma = \{L(A) \twoheadrightarrow L(B), L(B) \to L(A, C)\}$. The only minimal key is $L(B)$. From $L(A) \twoheadrightarrow L(B)$ and the Brouwerian-complement rule follows that $L(A) \twoheadrightarrow L(A, C)$ is in Σ' and therefore also in Σ^+. The instance $r = \{(a, b_1, c), (a, b_2, c)\}$ with distinct $b_1, b_2 \in dom(B)$ satisfies Σ and the projections of both elements on $L(A, C)$ are identical. It follows that N is type-2 and type-3 redundant with respect to Σ. However, N is not type-1 redundant as every dependency in Σ contains the subattribute $L(B)$ and no instance over N can have duplicates on a dependency in Σ.

3.2 The Proposal

Fourth normal form (4NF,[14]) has been introduced as an extension of Boyce-Codd normal form (BCNF) and has been intensely studied. A relation schema R is in 4NF with respect to a set Σ of FDs and MVDs defined on R if and only if every $X \twoheadrightarrow Y \in \Sigma^*$ is trivial or X is a superkey for R with respect to Σ.

Definition 5. Let Σ be a set of FDs and MVDs on the nested attribute N. We say that N is in *Nested List Normal Form (NLNF)* with respect to Σ if and only if every $X \twoheadrightarrow Y \in \Sigma^+$ is an inevitable dependency on N with respect to Σ or X is a superkey for N with respect to Σ. □

Note that NLNF generalises 4NF from relational databases. In fact, every inevitable dependency on the record-valued attribute $R(A_1, \ldots, A_n)$ must be trivial since the join-irreducibles of $R(A_1, \ldots, A_n)$ form an anti-chain with respect to $\leq$.

One may define N to be in *Nested List Fourth Normal Form (NL4NF)* with respect to Σ if and only if every $X \twoheadrightarrow Y \in \Sigma^+$ is a trivial dependency on N with respect to Σ or X is a superkey for N with respect to Σ. In this case, NL4NF also extends 4NF from relational databases and implies NLNF as every trivial dependency is also inevitable. However, NLNF is strictly weaker than NL4NF as there are, in general, inevitable dependencies which are not trivial. NLNF for FDs and MVDs subsumes the NLNF for FDs only [21].

HALFTONING(Brightness,INPUT[Level],OUTPUT[Bit]) is not in NLNF with respect to the set Σ of dependencies previously specified. The MVD HALFTONING(Brightness,INPUT[λ]) $\twoheadrightarrow$ HALFTONING(INPUT[Level]) is neither inevitable nor is HALFTONING(Brightness,INPUT[λ]) a superkey with respect to Σ.

3.3 Characterising NLNF

Given some nested attribute N and some set Σ of FDs and MVDs on N, how can we verify that N is in NLNF? By Definition 5 one needs to inspect every

$X \twoheadrightarrow Y$ derivable from Σ, i.e., every $X \twoheadrightarrow Y$ in Σ^+ must be inevitable or X must be a superkey. However, an inspection of every dependency in Σ suffices.

Theorem 4. *Let Σ be a set of FDs and MVDs on the nested attribute N. N is in NLNF with respect to Σ if and only if for every $X \rightarrow Y \in \Sigma$ or $X \twoheadrightarrow Y \in \Sigma$ which is not inevitable on N with respect to Σ, the left-hand side X is a superkey for N with respect to Σ.*

Proof (Sketch). One can show the following result using the inference rules from Theorem 3. For any $X \twoheadrightarrow W$ or $X \rightarrow W$ in Σ^+ which is not inevitable on N with respect to Σ, there is some $X' \twoheadrightarrow Y$ or $X' \rightarrow Y$ in Σ with $X' \leq X \sqcup \bigsqcup\{Z \sqcap Z^{\mathcal{C}} \mid X \twoheadrightarrow Z \in \Sigma^+\}$ which is not inevitable on N with respect to Σ.

Suppose N is not in NLNF with respect to Σ. Then there is some $X \twoheadrightarrow Y \in \Sigma^+$ which is not inevitable and where X is not a superkey for N with respect to Σ. The result above shows that there is some $X' \rightarrow Y'$ or $X' \twoheadrightarrow Y'$ in Σ which is not inevitable and where $X' \leq X \sqcup \bigsqcup\{Z \sqcap Z^{\mathcal{C}} \mid X \twoheadrightarrow Z \in \Sigma^+\}$ holds. The mixed meet rule guarantees that $X' \leq X^+$ and therefore $X \rightarrow X' \in \Sigma^+$. Since X is not a superkey for N with respect Σ, neither can X' be.

The remaining direction is a consequence of $\Sigma \subseteq \Sigma^+$ and the implication rule. □

We will now give another characterisation of NLNF. The result extends a classical result for relational databases [15]. In order to verify whether an instance over N in NLNF satisfies all dependencies it is sufficient to verify that all key dependencies and all inevitable dependencies are satisfied. Unlike the relational case where it is enough to look at all key dependencies for a relation schema in 4NF, one still needs to deal with all inevitable dependencies that are not trivial when a nested attribute in NLNF is given.

Theorem 5. *Let N be a nested attribute and Σ a set of FDs and MVDs on N. N is in NLNF with respect to Σ if and only if every $r \subseteq dom(N)$ with $\models_r \Sigma_{key} \cup \Sigma^+_{inev}$ implies $\models_r \Sigma$.*

Proof (Sketch). It is not difficult to see that the existence of some $r \subseteq dom(N)$ with $\models_r \Sigma_{\text{key}} \cup \Sigma^+_{\text{inev}}$ and $\not\models_r \Sigma$ implies the violation of the NLNF condition.

Suppose N is not in NLNF with respect to Σ. Let $X \twoheadrightarrow Y \in \Sigma^+$ be not inevitable and X not be a superkey for N with respect to Σ. One can show that there is some $r \subseteq dom(N)$ with $\models_r \Sigma_{\text{key}} \cup \Sigma^+_{\text{inev}}$ and $\not\models_r \Sigma$. In fact, one defines $X^+_{\text{inev}} = \bigsqcup\{Z \mid X \rightarrow Z \in \Sigma^+_{\text{inev}}\}$ and chooses $r \subseteq dom(N)$ with $r = \{t, t'\}$ such that

$$\pi^N_W(t) = \pi^N_W(t') \qquad \text{if and only if} \qquad W \leq X^+_{\text{inev}}.$$

Note that such t, t' can always be constructed [22, Lemma 3.2]. □

3.4 Type-2 and Type-3 Redundancy

We have seen that HALFTONING(Brightness,INPUT[Level],OUTPUT[Bit]) is not in NLNF. So far, this means only that the schema does not satisfy a syntactic condition with respect to the given set of dependencies. In this section we show the equivalence of NLNF to several semantic design desiderata. This will reveal what NLNF actually achieves.

The first semantic justification of Nested List Normal Form is that a nested attribute N is in NLNF with respect to a given set Σ of FDs and MVDs precisely if N is not type-3 redundant with respect to Σ.

Theorem 6. *Let Σ be a set of FDs and MVDs on the nested attribute N. Then is N in NLNF with respect to Σ if and only if N is not type-3 redundant with respect to Σ.*

Proof (Sketch). It is not difficult too see that type-3 redundancy implies the violation of the NLNF condition. Suppose N is not type-3 redundant with respect to Σ, and $X \twoheadrightarrow Y \in \Sigma^+$ is not inevitable with respect to Σ. We sketch that X is a superkey for N with respect to Σ. As N is not type-3 redundant we have $t_1 = t_2$ for all $t_1, t_2 \in r \subseteq dom(N)$ with $\models_r \Sigma$ and $\pi^N_{X \sqcup Y}(t_1) = \pi^N_{X \sqcup Y}(t_2)$. It follows that $X \sqcup Y$ is a superkey for N with respect to Σ. One can show that $X \rightarrow Y^{\mathcal{C}} \in \Sigma^+$, and $X \rightarrow Y^{\mathcal{C}}$ is not inevitable with respect to Σ. It follows that $X \sqcup Y^{\mathcal{C}}$ is a superkey for N with respect to Σ. Otherwise it is possible to construct some $r \subseteq dom(N), \mid r \mid \geq 2$ with $\models_r \Sigma$ and for all distinct $t_1, t_2 \in r$ we have $\pi^N_{X \sqcup Y^{\mathcal{C}}}(t_1) = \pi^N_{X \sqcup Y^{\mathcal{C}}}(t_2)$ contradicting the fact that N is not type-3 redundant with respect to Σ. Consequently, X is a superkey for N with respect to Σ. □

The two notions of type-2 and type-3 redundancy coincide.

Theorem 7. *Let Σ be a set of FDs and MVDs on the nested attribute N. Then is N type-2 redundant with respect to Σ if and only if N is type-3 redundant with respect to Σ.*

Proof (Sketch). Type-2 redundancy implies type-3 redundancy as $\Sigma' \subseteq \Sigma^+$. We sketch that if N is not type-2 redundant, then it is also not type-3 redundant. If $X \twoheadrightarrow Y$ or $X \rightarrow Y \in \Sigma'$ is not inevitable, then one can show that $X \sqcup Y$ is a superkey for N with respect to Σ. Consider every dependency in Σ' that is not inevitable with respect to Σ:

- for an FD $X \rightarrow Y$ we infer that X must be a superkey for N with respect to Σ,
- for an MVD $X \twoheadrightarrow Y$ we also have $X \twoheadrightarrow Y^{\mathcal{C}} \in \Sigma'$. Consequently, both $X \sqcup Y$ and $X \sqcup Y^{\mathcal{C}}$ are superkeys for N with respect to Σ, and one can show that both $X \rightarrow Y^{\mathcal{C}}, X \rightarrow Y^{\mathcal{CC}} \in \Sigma^+$. That gives $X \rightarrow N \in \Sigma^+$, i.e., X is a superkey for N with respect to Σ.

Since $\Sigma \subseteq \Sigma'$ holds, the left-hand side of every dependency in Σ which is not inevitable with respect to Σ is a superkey for N. Theorem 4 shows that N is in NLNF with respect to Σ and we conclude that N is not type-3 redundant with respect to Σ by Theorem 6. □

It follows that the notion of type-2 redundancy is invariant under different choices of equivalent sets of FDs and MVDs.

Corollary 1. *Let $N \in \mathcal{N}A$, and Σ and Θ two equivalent sets of FDs and MVDs on N. Then N is type-2 redundant with respect to Σ if and only if N is type-2 redundant with respect to Θ.* □

3.5 Type-1 Redundancy

Unlike type-2 and type-3 redundancy, type-1 redundancy does depend on the choice of equivalent sets of dependencies. We will now characterise type-1 redundancy syntactically.

Theorem 8. *Let Σ be a set of FDs and MVDs on the nested attribute N. Then the following conditions are equivalent:*

1. *N is not type-1 redundant with respect to Σ,*
2. *for every $X \twoheadrightarrow Y$ and $X \rightarrow Y$ in Σ which is not inevitable on N with respect to Σ the subattribute $X \sqcup Y$ is a superkey for N with respect to Σ, and*
3. *for every $X \twoheadrightarrow Y$ and $X \rightarrow Y$ in Σ which is not inevitable on N with respect to Σ we have $X \rightarrow Y^{\mathcal{C}} \in \Sigma^+$.* □

Let $N = L(A,B,C), \Sigma = \{L(A) \twoheadrightarrow L(B), L(B) \rightarrow L(A,C)\}$, and $\Theta = \{L(A) \twoheadrightarrow L(C), L(B) \rightarrow L(A,C)\}$. $L(B)$ is the only minimal key with respect to Σ and Θ, and Σ and Θ are equivalent sets of FDs and MVDs. N is not type-1 redundant with respect to Σ since every $X \twoheadrightarrow Y$ and every $X \rightarrow Y$ in Σ satisfies $L(B) \leq X \sqcup Y$. However, N is type-1 redundant with respect to Θ since $A \sqcup C$ is not a superkey on N with respect to Σ.

3.6 Pure MVDs

We will now provide a sufficient condition under which the different types of redundancies are equivalent. Let Σ be a set of FDs and MVDs defined on the nested attribute N. An MVD $X \twoheadrightarrow Y \in \Sigma$ is called *pure* if and only if neither $X \rightarrow Y$ nor $X \rightarrow Y^{\mathcal{C}}$ are in Σ^+. Pure MVDs are not inevitable on N with respect to Σ. The set Σ is called pure if and only if every MVD in Σ is pure.

Pure MVDs reflect pure multivalued information and can therefore not be captured by FDs. A set Σ of MVDs and FDs contains at least one pure MVD if and only if Σ is not equivalent to a set of FDs.

Theorem 9. *If $N \in \mathcal{N}A$ and Σ is a pure set of FDs and MVDs on N, then N is type-1 redundant if and only if N is type-3 redundant.* □

Corollary 2. *Let $N \in \mathcal{N}A$, and Σ be a pure set of FDs and MVDs on N. If N is type-1 redundant with respect to Σ, then is N type-1 redundant with respect to any set Θ of FDs and MVDs on N that is equivalent to Σ.* □

3.7 Value Redundancy

The previously introduced notions of redundancy have one major deficiency in common. They all depend on the syntactic structure of FDs and MVDs making it difficult to further generalise those notions to other types of dependencies or adapting those definitions to other data models. In relational databases the notion of value redundancy has been introduced to overcome those deficiencies [35]. The occurrence of some flat attribute value in some flat relation is redundant if it can be derived from other data values in that relation and the set of dependencies which apply to that relation. A relation schema R is redundant if there exists a legal R-relation which contains an occurrence of a flat attribute value such that *any* change to this occurrence results in the violation of at least one dependency.

Care must be taken when this notion of value-redundancy is generalised to the presence of lists. Consider for instance the list-valued nested attribute OUTPUT[Bit] together with the element $[0, 1, 1, 1]$ which has projection $[ok, ok, ok, ok]$ on the subattribute OUTPUT[λ]. Then, the value occurrence $[ok, ok, ok, ok]$ cannot be changed without affecting the list $[0, 1, 1, 1]$. On the other hand, arbitrary changes to $[0, 1, 1, 1]$ may also affect the projection $[ok, ok, ok, ok]$. An alteration of the number of the elements in $[0, 1, 1, 1]$, e.g. removing the last two elements results in $[0, 1]$, and also results in a different projection, $[ok, ok]$ in this case.

In general, changes to the value $\pi_Y^N(t)$ on a join-irreducible Y will cause a change of values $\pi_Z^N(t)$ on any join-irreducible Z with $Y \leq Z$. It is therefore advisable to consider only value occurrences $\pi_M^N(t)$ on maximal join-irreducibles $M \in \mathcal{B}_{\mathcal{M}}(N)$. Thus, the role that flat attributes played in the definition of value redundancy in relational databases, is now taken by maximal join-irreducibles in the context of lists. Moreover, we will not consider arbitrary changes to $\pi_M^N(t)$, but only those which do not cause any changes to $\pi_W^N(t)$ for all $W < M$. We call the replacement of some data value $\pi_M^N(t)$ by $m \in dom(M)$ *admissible* if and only if $\pi_W^N(t) = \pi_W^M(m)$ for all $W < M$.

Definition 6. Let $N \in \mathcal{N}\mathrm{A}$, $M \in \mathcal{B}_{\mathcal{M}}(N)$, Σ a set of dependencies on N, $r \subseteq dom(N)$ and $t \in r$. The data value occurrence $\pi_M^N(t)$ is *redundant* if and only if for every admissible replacement of $\pi_M^N(t)$ by a value m with $\pi_M^N(t) \neq m$ that results in the modified instance $r' \subseteq dom(N)$ we have $\not\models_{r'} \Sigma$. □

The last definition extends the notion of value-redundancy from the relational case [35]. In the presence of records only, the join-irreducibles form an anti-chain with respect to Σ and, consequently, every join-irreducible is maximal and all replacements are admissible. Consider the following legal snapshot of 6 tuples

$$(\tfrac{1}{2}, [0, \tfrac{1}{2}], [0, 0]),\ (\tfrac{1}{2}, [0, \tfrac{1}{2}], [0, 1]),\ (\tfrac{1}{2}, [0, \tfrac{1}{2}], [1, 0])$$
$$(\tfrac{1}{2}, [\tfrac{1}{2}, 0], [0, 0]),\ (\tfrac{1}{2}, [\tfrac{1}{2}, 0], [0, 1]),\ (\tfrac{1}{2}, [\tfrac{1}{2}, 0], [1, 0])$$

over HALFTONING(Brightness,INPUT[Level],OUTPUT[Bit]). The admissible replacements of $[0, 0]$ are $[0, 1], [1, 0], [1, 1]$. The data value occurrence of any of these 6 tuples projected on HALFTONING(INPUT[Level]) is redundant as every admissible replacement of such an occurrence leads to a violation of the MVD

$$\text{HALFTONING}(\text{Brightness},\text{INPUT}[\lambda]) \twoheadrightarrow \text{HALFTONING}(\text{INPUT}[\text{Level}]).$$

The same applies to the data value occurrences of any tuple projected on HALFTONING(OUTPUT[Bit]).

The nested attribute N is in Value Redundancy Free Normal Form (VRFNF) with respect to a set Σ of dependencies defined on N if and only if there does not exist $r \subseteq dom(N)$ with $\models_r \Sigma$ which contains a data value occurrence that is redundant.

Theorem 10. *Let Σ be a set of FDs and MVDs on the nested attribute N. Then N is in VRFNF with respect to Σ if and only if N is in NLNF with respect to Σ.*

Proof (Sketch). We sketch first that if N is not in VRFNF with respect to Σ, then N is also not in NLNF with respect to Σ. If N is not in VRFNF with respect to Σ, then there is some $r \subseteq dom(N)$, some $t \in r$ and some $M \in \mathcal{B}_{\mathcal{M}}(N)$ such that every admissible replacement of $\pi_M^N(t)$ results in a modified instance violating Σ. That is, if $\pi_M^N(t)$ is changed to a value m' such that $m' \notin \pi_M(r) = \{\pi_M^N(s) : s \in r\}$, resulting in the new element t' and the modified instance $r' = (r - \{t\}) \cup \{t'\}$, then $\not\models_{r'} \Sigma$. One can now show that a violation of some FD or MVD in Σ by r' implies a violation of the NLNF condition. We omit the details.

If N is not in NLNF with respect to Σ, then there is some $X \twoheadrightarrow Y \in \Sigma^+$ which is not inevitable on N with respect to Σ and where X is not a superkey for N with respect to Σ. In the following let $DepB(X)$ denote the dependency basis of X with respect to Σ [22]. Let $DepB(X) = \{W_{0,1}, \dots, W_{0,m}, W_1, \dots, W_k\}$ with $W_{0,i} \leq X^+$ and $W_j \not\leq X^+$ for $i = 1, \dots, m$ and $j = 1, \dots, k$. Since X is not a superkey for N with respect to Σ, there is some $W_j \in DepB(X)$ with $W_j \not\leq X^+ < N$. One can show that there is some $r = \{t, t'\} \subseteq dom(N)$ with $\models_r \Sigma$ and $\pi_W^N(t) = \pi_W^N(t')$ iff $W \leq X^+ \sqcup \bigsqcup\{W_n : n \neq j\}$. Suppose there is some $M \in \mathcal{B}_{\mathcal{M}}(N)$ with $M \not\leq X$, but $M \leq X^+$. Then $\pi_M^N(t)$ and $\pi_M^N(t')$ are both redundant: changing one of these values results in a violation of $X \to X^+ \in \Sigma^+$ which is not inevitable on N with respect to Σ.

Alternatively, $(X^+)^{\mathcal{CC}} \leq X$. Assume there is only one $W_j \in DepB(X)$ with $W_j \not\leq X^+$. $X \twoheadrightarrow Y \in \Sigma^+$ implies that Y is the join over some elements of $DepB(X)$ and as $X \twoheadrightarrow Y \in \Sigma^+$ is not inevitable on N with respect to Σ we have $Y^{\mathcal{CC}} \not\leq X$. This leaves us with $W_j \leq Y$ and thus $N = X \sqcup W_j \leq X \sqcup Y$, i.e., $N = X \sqcup Y$. This, however, is a contradiction since $X \twoheadrightarrow Y \in \Sigma^+$ is not inevitable on N with respect to Σ. We therefore have two distinct $W_i, W_j \in DepB(X)$. Consequently, for all $M \in \mathcal{B}_{\mathcal{M}}(N)$ with $M \leq W_i$ (at least one such M exists) the values $\pi_M^N(t)$ and $\pi_M^N(t')$ are redundant since changing any one of these values results in the violation of $X \twoheadrightarrow W_i \in \Sigma^+$. Therefore, N is not in VRFNF with respect to Σ. □

Since NLNF for FDs and MVDs reduces to NLNF for FDs if only FDs are present, Theorem 10 shows that NLNF for FDs is the exact condition required to avoid value redundancy when only FDs are present. This extends previous results [21] where value redundancy had not been considered at all.

It also follows that value redundancy is equivalent to type-2 and type-3 redundancy, and equivalent to type-1, type-2 and type-3 redundancy whenever Σ is a pure set of FDs and MVDs.

3.8 Strong Update Anomalies

In the relational model of data a relation schema in 4NF does not have any update anomalies. This is another justification why relation schemata should be in 4NF [35]. The next example reveals a surprising fact. Consider

$$\text{Halftoning}(\text{Brightness},\text{Input}[\lambda],\text{Output}[\text{Bit}])$$

which is in NLNF with respect to the FD

$$\text{Halftoning}(\text{Input}[\lambda]) \rightarrow \text{Halftoning}(\text{Output}[\lambda])$$

and therefore free of any form of redundancies. Say a simple database consists of the single element $(\frac{1}{2}, [ok, ok], [0, 1])$ and the element $(\frac{1}{2}, [ok, ok], [0, 0, 0, 0])$ happens to be inserted. Then all key dependencies are trivially satisfied by the new relation, but the FD

$$\text{Halftoning}(\text{Input}[\lambda]) \rightarrow \text{Halftoning}(\text{Output}[\lambda])$$

is violated. This example shows that, in general, the absence of redundancy for a nested attribute does not imply the absence of insertion anomalies. Therefore, it cannot be expected that nested attributes in NLNF do not have update anomalies. We define, however, strong update anomalies in the context of nested attributes. The main difference to the relational case is that updated relations which define any strong anomaly do not only satisfy all key dependencies on the nested attribute, but also all inevitable dependencies.

Definition 7. Let Σ be a set of FDs and MVDs on the nested attribute N.

1. We say that N has a *strong insertion anomaly* with respect to Σ if and only if there is some $r \subseteq dom(N)$ with $\models_r \Sigma$ and some $t \notin r$ with $\models_{r\cup\{t\}} \Sigma_{\text{key}} \cup \Sigma^+_{\text{inev}}$, but $\not\models_{r\cup\{t\}} \Sigma$.
2. We say that N has a *strong deletion anomaly* with respect to Σ if and only if there is some $r \subseteq dom(N)$ with $\models_r \Sigma$ and some $t \in r$ with $\models_{r-\{t\}} \Sigma_{\text{key}} \cup \Sigma^+_{\text{inev}}$, but $\not\models_{r-\{t\}} \Sigma$.
3. We say that N has a *strong replacement anomaly*
   - *of type 1* with respect to Σ if and only if there is some $r \subseteq dom(N)$ with $\models_r \Sigma$ and some $t \in r$ and $t' \in dom(N)$ with $\pi^N_K(t) = \pi^N_K(t')$ for some minimal key K on N and $\models_{r-\{t\}\cup\{t'\}} \Sigma_{\text{key}} \cup \Sigma^+_{\text{inev}}$ and $\not\models_{r-\{t\}\cup\{t'\}} \Sigma$ hold.
   - *of type 2* with respect to Σ if and only if there is some $r \subseteq dom(N)$ with $\models_r \Sigma$ and some $t \in r$ and $t' \in dom(N)$ with $\pi^N_K(t) = \pi^N_K(t')$ for some distinguished minimal key K on N and $\models_{r-\{t\}\cup\{t'\}} \Sigma_{\text{key}} \cup \Sigma^+_{\text{inev}}$ and $\not\models_{r-\{t\}\cup\{t'\}} \Sigma$ hold.

– *of type 3* with respect to Σ if and only if there is some $r \subseteq dom(N)$ with $\models_r \Sigma$ and some $t \in r$ and $t' \in dom(N)$ with $\pi_K^N(t) = \pi_K^N(t')$ for all minimal keys K on N and $\models_{r-\{t\}\cup\{t'\}} \Sigma_{\text{key}} \cup \Sigma_{\text{inev}}^+$ and $\not\models_{r-\{t\}\cup\{t'\}} \Sigma$ hold. □

If updates only alter values on maximal join-irreducibles and keep values on non-maximal join-irreducibles fixed, then only key dependencies need to be checked. Otherwise, one also needs to check all inevitable dependencies that are not trivial. The next theorem generalises a well-known result from relational databases [15].

Theorem 11. *Let Σ be a set of FDs and MVDs on the nested attribute N. Then is N in NLNF with respect to Σ if and only if N does not have any strong insertion anomaly with respect to Σ.* □

While it is relatively easy to see that a nested attribute in NLNF does not have any strong update anomalies, the absence of a strong update anomaly does not necessarily imply NLNF. For strong deletion anomalies we obtain the following result.

Theorem 12. *Let Σ be a set of FDs and MVDs on the nested attribute N. N does not have any strong deletion anomaly with respect to Σ if and only if Σ is equivalent to a set of FDs.*

Proof (Sketch). If Σ is equivalent to a set $\Sigma_{\mathcal{F}}$ of FDs, then no deletion anomaly with respect to $\Sigma_{\mathcal{F}}$ can occur. Since strong deletion anomalies are invariant under different choices of covers, it follows that no deletion anomaly with respect to Σ can occur. Hence, N does not have any strong deletion anomaly with respect to Σ.

If N is not equivalent to a set of FDs one can show that there is some pure MVD $X \twoheadrightarrow Y$ in Σ. Then there is some $r \subseteq dom(N)$ with $\models_r \Sigma$ and four distinct elements $t_1, t_2, t_3, t_4 \in r$ with $\pi_X^N(t_1) = \pi_X^N(t_2) = \pi_X^N(t_3) = \pi_X^N(t_4)$, $\pi_{W_1}^N(t_1) = \pi_{W_1}^N(t_3)$, $\pi_{W_1}^N(t_2) = \pi_{W_1}^N(t_4)$, $\pi_{W_2}^N(t_1) = \pi_{W_2}^N(t_4)$ and $\pi_{W_2}^N(t_2) = \pi_{W_2}^N(t_3)$ such that $W_1, W_2 \in DepB(X)$ are distinct and $W_i \not\leq X^+$ for $i = 1, 2$. However, r has a deletion anomaly since deleting any of the four elements from r results in a violation of $X \twoheadrightarrow W_i \in \Sigma^+$ which is not inevitable on N with respect to Σ and where X is not a superkey for N with respect to Σ. Consequently, there is some $X' \twoheadrightarrow Y'$ in Σ which is not inevitable and where $X' \leq X \sqcup \bigsqcup\{Z \sqcap Z^{\mathcal{C}} \mid X \twoheadrightarrow Z \in \Sigma^+\}$ holds. The mixed meet rule guarantees that $X' \leq X^+$ and therefore $X \to X' \in \Sigma^+$. Since X is not a superkey for N with respect to Σ, neither can X' be. □

It is the subject of future research to study the relationship between the various forms of strong replacement anomalies and NLNF. We conjecture that the results for strong replacement anomalies will be similar to those established for 4NF and key-based (fact-based) replacement anomalies in case of relational databases [35].

4 NLNF Decomposition

So far we have proposed the Nested List Normal Form as a desirable normal form that we aim to achieve in a database. We now tackle the problem of how to obtain

NLNF. Theorem 2 indicates that an extension of the relational decomposition approach [12, 17] can be applied to NLNF. Given some nested attribute N and a set Σ of FDs and MVDs defined on N, the decomposition approach aims at finding a set of subattributes of N each of which is in NLNF with respect to the corresponding set of all implied FDs and MVDs on that subattribute. Moreover, any instance of N that satisfies Σ is the generalised natural join of all its projections on the subattributes, i.e., every valid database on N can be decomposed without loss of information.

Definition 8. Let $N \in \mathcal{N}A$, $N_1, \ldots, N_k \in Sub(N)$, and Σ a set of FDs and MVDs defined on N. The set $\{N_1, \ldots, N_k\}$ is called a *lossless join decomposition of* N *with respect to* Σ if and only if $N = \bigsqcup\{N_1, \ldots, N_k\}$ and $r = \pi_{N_1}(r) \bowtie \cdots \bowtie \pi_{N_k}(r)$ holds for all $r \subseteq dom(N)$ with $\models_r \Sigma$. The set $\{N_1, \ldots, N_k\}$ is a *lossless NLNF (NL4NF) decomposition of* N *with respect to* Σ if and only if $\{N_1, \ldots, N_k\}$ is a lossless join decomposition of N with respect to Σ and N_i is in NLNF (NL4NF) with respect to $\pi_{N_i}(\Sigma^+)$ for every $i = 1, \ldots, k$, and where $\pi_M(\Sigma) = \{X \to Y \in \Sigma \mid X \sqcup Y \leq M\} \cup \{X \twoheadrightarrow Y \sqcap M \in \Sigma \mid X \leq M\}$. □

We will now show that it is possible to obtain a lossless NLNF decomposition for any given nested attribute N and any given set of FDs and MVDs on N. Whenever an MVD in the current state of the output schema violates NLNF, the decomposition algorithm removes the cause for this violation of NLNF by replacing the offending parent subattribute by two of its proper child subattributes which can be joined losslessly to reconstruct their parent.

Algorithm 1 (Lossless NLNF decomposition)

Input: $\boldsymbol{N} \in \boldsymbol{\mathcal{N}}$A, set $\boldsymbol{\Sigma}$ of FDs and MVDs on $\boldsymbol{N}$

Output: set $\boldsymbol{\mathcal{S} = \{(N_1, \Sigma_1), \ldots, (N_k, \Sigma_k)\}}$ where $\boldsymbol{\Sigma_i}$ is set of FDs and MVDs on $\boldsymbol{N_i \in Sub(N)}$ and $\boldsymbol{\{N_1, \ldots, N_k\}}$ is lossless NLNF decomposition of $\boldsymbol{N}$ with respect to $\boldsymbol{\Sigma}$

Method:

VAR $X, Y, N_1, N_2 \in Sub(N)$
DECOMPOSE(N,Σ)
BEGIN
  IF N in NLNF wrt Σ, **THEN** $\mathcal{S} := \{(N, \Sigma)\}$;
  ELSE
    LET $X \twoheadrightarrow Y \in \Sigma$ be not inevitable on N wrt Σ and $\Sigma \not\models X \to N$;
    $N_1 := X \sqcup Y$;
    $N_2 := X \sqcup Y^{\mathcal{C}}$;
    $\mathcal{S} :=$ **DECOMPOSE**$(N_1, \pi_{N_1}(\Sigma^+))\cup$**DECOMPOSE**$(N_2, \pi_{N_2}(\Sigma^+))$;
  ENDIF;
  RETURN($\mathcal{S}$);
END; □

Theorem 13. *Algorithm 1 is correct.* □

For relational databases it is well-known that any relation schema with any set of FDs and MVDs defined on it, can be decomposed into subschemata that are all in 4NF with respect to the projected sets of FDs and MVDs. In the presence of lists, however, the situation is different. The next result is further evidence that a simple extension of 4NF to NL4NF is too strong.

Theorem 14. *There are nested attributes N and sets Σ of FDs and MVDs on N for which no lossless NL4NF-decomposition exists.*

Proof. Let $N = L[A]$ and $\Sigma = \{\lambda \twoheadrightarrow L[\lambda]\}$. The MVD is not trivial and λ is not a superkey for N with respect to Σ. Consequently, N is not in NL4NF with respect to Σ. However, any decomposition of $L[A]$ must contain the nested attribute $L[A]$ itself. Therefore, no lossless NL4NF decomposition of $L[A]$ with respect to Σ exists. □

Algorithm 1 generalises the well-known 4NF decomposition algorithm for relational databases, see for instance [27–p.270]. It follows that the NLNF decomposition algorithm causes at least as many computational problems as its relational counterpart (,e.g. running time and dependency-preservation). However, the problems do not become harder in the presence of lists. Due to lack of space we cannot go into further details. The 4NF-decomposition from [17] may indicate how to improve the NLNF decomposition according to its running time.

We continue the example from digital halftoning. The MVD

$$\text{Halftoning}(\text{Brightness},\text{Input}[\lambda]) \twoheadrightarrow \text{Halftoning}(\text{Input}[\text{Level}])$$

is neither inevitable nor is Halftoning(Brightness,Input[λ]) a superkey. A first decomposition yields

$$N_1{=}\text{Halftoning}(\text{Brightness},\text{Input}[\text{Level}],\lambda) \text{ and}$$
$$N_2'{=}\text{Halftoning}(\text{Brightness},\text{Input}[\lambda],\text{Output}[\text{Bit}]).$$

The nested attribute N_1 is in NLNF with respect to $\pi_{N_1}(\Sigma)$. The attribute N_2' carries the inevitable FD Halftoning(Input[λ]) $\rightarrow$ Halftoning(Output[λ]), and the FD Halftoning(Output[λ]) $\rightarrow$ Halftoning(Input[λ]) which is not inevitable on N_2'. A further decomposition of N_2' gives

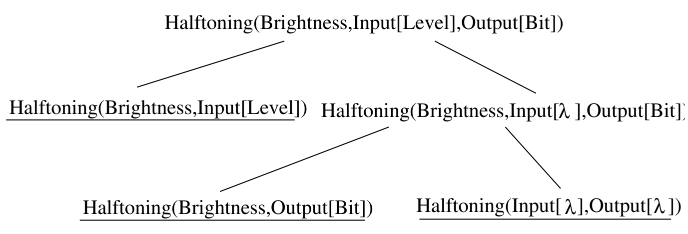

Fig. 2. Decomposition Tree for the Halftoning Example

N_2=HALFTONING(Brightness,λ,OUTPUT[Bit]) and
N_3=HALFTONING(λ,INPUT[λ],OUTPUT[λ])

which are in NLNF with respect to $\pi_{N_2}(\Sigma)$ and $\pi_{N_3}(\Sigma)$, respectively.

Since HALFTONING(Brightness,INPUT[Level],OUTPUT[Bit]) is not in NLNF with respect to Σ, and Σ is pure, the schema is redundant in every sense of notions that have been provided here. It also means that the schema carries strong insertion and deletion anomalies. The decomposition approach suggests to store the information in three separate schemata N_1, N_2 and N_3 which are all in NLNF with respect to the projected sets of FDs and MVDs.

5 Related and Future Work

We have proposed a suitable extension of 4NF from relational databases to the presence of lists. The results demonstrate that 4NF together with all its beautiful semantic properties can be generalised to arbitrary finite nesting of records and lists (slightly extended framework of DTDs in which only concatenation and Kleene closure are allowed). In particular, NLNF was semantically justified in several ways by showing the equivalence to the absence of appropriate extensions of different notions of redundancy and insertion anomalies. The data model has put emphasis on the record and list constructor, but will be extended to other data constructors in the future. The feature of lists to model order makes it possible to focus on join-irreducible subattributes and makes it possible to extend well-known results allowing to capture new application domains. Since set and multiset constructor neglect the order of their elements an extension of these results will be challenging.

The minimal axiomatisation from Theorem 3 follows directly from previous work [20, 22]. The complete set of inference rules is used in several proof arguments. NLNF for FDs themselves has been studied previously [21]. While [3] define information-theoretic measures to address the problem of well-designed data in data formats different from the relational data model (such as XML) we have offered an algebraic approach to database design which classifies data models according to the data type constructors supported.

Unlike our work, none of the XML FDs and MVDs [2, 37] take list equality into consideration. We believe that list equality is natural and common in real applications and should be included in defining data dependencies. An alternative way to define object equality in the context of XML can be based on homomorphisms in XML graphs [19, 23]. Notably both [18, 31] have considered set equality in their definitions of FDs in the nested relational data model.

References

1. S. Abiteboul, R. Hull, and V. Vianu. *Foundations of Databases.* Addison-Wesley, 1995.
2. M. Arenas and L. Libkin. A normal form for XML documents. *TODS*, 29(1):195–232, 2004.

3. M. Arenas and L. Libkin. An information-theoretic approach to normal forms for relational and XML data. *J.ACM*, 52(2):246–283, 2005.
4. T. Asano, N. Katoh, K. Obokata, and T. Tokuyama. Matrix rounding under the L_p-discrepancy measure and its application to digital halftoning. *SIAM Journal on Computing*, 32(6):1423–1435, 2003.
5. M. Atkinson, F. Bancilhon, D. DeWitt, K. Dittrich, D. Maier, and S. Zdonik. The object-oriented database system manifesto. In *Proceedings of the International Conference on Deductive and Object-Oriented Databases*, pages 40–57, 1989.
6. C. Beeri, P. A. Bernstein, and N. Goodman. A sophisticate's introduction to database normalization theory. In *VLDB*, pages 113–124, 1978.
7. P. A. Bernstein and N. Goodman. What does Boyce-Codd normal form do? In *VLDB*, pages 245–259, 1980.
8. J. Biskup. Database schema design theory: achievements and challenges. In *Information Systems and Data Management*, number 1066 in LNCS, pages 14–44. Springer, 1995.
9. J. Biskup. Achievements of relational database schema design theory revisited. In *Semantics in databases*, number 1358 in LNCS, pages 29–54. Springer, 1998.
10. T. Bray, J. Paoli, C. M. Sperberg-McQueen, E. Maler, and F. Yergeau. Extensible markup language (XML) 1.0 (third edition) W3C recommendation 04 February 2004. http://www.w3.org/TR/2004/REC-xml-20040204/, 2004.
11. P. P. Chen. The entity-relationship model: Towards a unified view of data. *TODS*, 1:9–36, 1976.
12. E. F. Codd. Further normalization of the database relational model. In *Courant Computer Science Symposia 6: Data Base Systems*, pages 33–64. Prentice-Hall, 1972.
13. E. F. Codd. Recent investigations in relational database system. In *Proceedings of the IFIP Conference*, pages 1017–1021, 1974.
14. R. Fagin. Multivalued dependencies and a new normal form for relational databases. *TODS*, 2(3):262–278, 1977.
15. R. Fagin. A normal form for relational databases that is based on domains and keys. *TODS*, 6(3):387–415, 1981.
16. R. W. Floyd and L. Steinberg. An adaptive algorithm for spatial grey scale. In *Proceedings of the Society of Information Display*, pages 75–77, 1976.
17. G. Grahne and K. P. Räihä. Database decomposition into 4NF. In *VLDB*, pages 186–196, 1983.
18. C. S. Hara and S. B. Davidson. Reasoning about nested functional dependencies. In *PODS*, pages 91–100, 1999.
19. S. Hartmann and S. Link. More functional dependencies for XML. In *ADBIS*, number 2798 in LNCS, pages 355–369. Springer, 2003.
20. S. Hartmann and S. Link. Multi-valued dependencies in the presence of lists. In *PODS*, pages 330–341, 2004.
21. S. Hartmann and S. Link. Normalisation in the presence of lists. In *ADC*, volume 27 of *CRPIT*, pages 53–64, 2004.
22. S. Hartmann, S. Link, and K.-D. Schewe. Functional and multivalued dependencies in nested databases generated by record and list constructor. accepted for Annals of Mathematics and Artificial Intelligence, 2006.
23. S. Hartmann and T. Trinh. Axiomatising functional dependencies for XML with frequencies. In *FoIKS*, this volume of LNCS. Springer, 2006.
24. R. Hull and R. King. Semantic database modeling: Survey, applications and research issues. *ACM Computing Surveys*, 19(3), 1987.

25. J. F. Jarvis, C. N. Judice, and W. H. Ninke. A survey of techniques for the display of continuous-tone pictures on bilevel display. *Computer Graphics Image Processing*, 5:13–40, 1976.
26. I. Katsavounidis and C.-C. J. Kuo. A multiscale error diffusion technique for digital halftoning. *Transactions on Image Processing*, 6(3):483–490, 1997.
27. M. Levene and G. Loizou. *A Guided Tour of relational databases and beyond*. Springer, 1999.
28. J. Li, S. Ng, and L. Wong. Bioinformatics adventures in database research. In *ICDT*, number 2572 in LNCS, pages 31–46. Springer, 2002.
29. J. C. C. McKinsey and A. Tarski. On closed elements in closure algebras. *Annals of Mathematics*, 47:122–146, 1946.
30. W. Y. Mok. A comparative study of various nested normal forms. *IEEE Transactions on Knowledge & Data Engineering*, 14(2):369–385, 2002.
31. M. A. Roth, H. F. Korth, and A. Silberschatz. Extended algebra and calculus for nested relational databases. *TODS*, 13(4):389–417, 1988.
32. D. Suciu. On database theory and XML. *SIGMOD Record*, 30(3):39–45, 2001.
33. B. Thalheim. *Entity-Relationship Modeling: Foundations of Database Technology*. Springer, 2000.
34. V. Vianu. A web odyssey: from Codd to XML. In *PODS*, pages 1–15, 2001.
35. M. Vincent. Semantic foundation of 4NF in relational database design. *Acta Informatica*, 36:1–41, 1999.
36. M. Vincent and J. Liu. Functional dependencies for XML. In *APWEB*, number 2642 in LNCS, pages 22–34. Springer, 2003.
37. M. Vincent, J. Liu, and C. Liu. A redundancy free 4NF for XML. In *XML Database Symposium*, number 2824 in LNCS, pages 254–266. Springer, 2003.