Acta Cybernetica **18** (2008) 529–556.

Keys and Armstrong Databases in Trees with Restructuring

Attila Sali* and Klaus-Dieter Schewe†

Abstract

The definition of keys, antikeys, Armstrong-instances are extended to complex values in the presence of several constructors. These include tuple, list, set and a union constructor. Nested data structures are built using the various constructors in a tree-like fashion. The union constructor complicates all results and proofs significantly. The reason for this is that it comes along with non-trivial restructuring rules. Also, so-called counter attributes need to be introduced. It is shown that keys can be identified with closed sets of subattributes under a certain closure operator. Minimal keys correspond to closed sets minimal under set-wise containment. The existence of Armstrong databases for given minimal key systems is investigated. A sufficient condition is given and some necessary conditions are also exhibited. Weak keys can be obtained if functional dependency is replaced by weak functional dependency in the definition. It is shown, that this leads to the same concept. Strong keys are defined as principal ideals in the subattribute lattice. Characterization of antikeys for strong keys is given. Some numerical necessary conditions for the existence of Armstrong databases in case of degenerate keys are shown. This leads to the theory of bounded domain attributes. The complexity of the problem is shown through several examples.

1 Introduction

The relational datamodel gave rise to theoretical research in several directions. Dependency structures were investigated as first-order logical sentences that are supposed to hold for all database instances [3]. On the other hand, their combinatorial investigations were fruitful resulting in nice problems, concepts, even as far topics as design and coding theory [8, 9, 12, 5].

The relational model has been extended or generalized to nested relational model [19], object oriented models [23], and object-relational models. The important structures of all these were captured by the higher-order Entity-Relationship

*Alfréd Rényi Institute of Mathematics, Hungarian Academy of Sciences, Budapest, P.O.B.127, H-1364 Hungary, E-mail: sali@renyi.hu

†Massey University, Information Science Research Centre & Department of Information Systems, Private Bag 11 222, Palmerston North, New Zealand, E-mail: k.d.schewe@massey.ac.nz

model [24, 25]. The semi-structured data and XML treated in [1] can also be considered as an object-oriented model.

The major new structure in all these models is the introduction of constructors that allow us to form complex data values from simpler ones. The dependencies of the relational model can be generalized to these higher-order models, and the axiomatization of certain dependencies was carried out in [13, 15, 16, 18, 14]. On the other hand, the induced combinatorial structures have not been investigated thoroughly yet. It is important from the point of view of schema design, to identify what kind of attributes can form *key systems*. The aim of the present paper is to continue the investigations started in [20, 21], thus generalizing the work of [6, 7, 8].

In Section 2 the necessary definitions are recalled. In Section 3 keys and antikeys are defined and it is shown that they correspond to *closed subsets* of the subattribute lattice under a certain closure operator. This is in sharp contrast with the Relational Datamodel (RDM), where *any* subset of the attributes could be a key for an appropriate system of functional dependencies. Section 4 deals with question of existence of Armstrong-instances. This question was first investigated by Armstrong [4] and Demetrovics [6] for functional dependencies in the RDM. Later Fagin [10] gave a necessary and sufficient condition for general dependencies in the relational context. Fagin's results are quite general, if types of dependencies are considered, however, they are only valid for relational databases, as the conditions he gave depend on direct products of relations.

In the present paper we treat functional dependencies in the higher order datamodel and a sufficient condition is given for the existence of Armstrong instance. In addition, we illustrate the complexity of the problem through several examples. Section 5 is devoted to strong keys, that are the closest analogs of keys in the RDM. Finally, Section 6 contains some inequalities of parameters that give necessary conditions for the existence of Armstrong-instances.

2 Preliminaries

In this section we define our model of nested attributes, which covers the gist of higher-order datamodels including XML. In particular, we investigate the structure of the set $\mathcal{S}(X)$ of subattributes of a given nested attribute X. We show that we obtain a non-distributive Brouwer algebra, i.e. a non-distributive lattice with relative pseudo-complements.

2.1 Nested Attributes

We start with a definition of simple attributes and values for them.

Definition 1. 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 *simple attributes*.

For the relational model a universe was enough, as a relation schema could be defined by a subset $R \subseteq \mathcal{U}$. For higher-order datamodels, however, we need nested attributes. In the following definition we use a set $\mathcal{L}$ of labels, and tacitly assume that the symbol λ is neither a simple attribute nor a label, i.e. $\lambda \notin \mathcal{U} \cup \mathcal{L}$, and that simple attributes and labels are pairwise different, i.e. $\mathcal{U} \cap \mathcal{L} = \emptyset$.

Definition 2. Let $\mathcal{U}$ be a universe and $\mathcal{L}$ a set of labels. The set $\mathcal{N}$ of *nested attributes* (over $\mathcal{U}$ and $\mathcal{L}$) is the smallest set with $\lambda \in \mathcal{N}$, $\mathcal{U} \subseteq \mathcal{N}$, and satisfying the following properties:

- for $X \in \mathcal{L}$ and $X'_1, \dots, X'_n \in \mathcal{N}$ we have $X(X'_1, \dots, X'_n) \in \mathcal{N}$;
- for $X \in \mathcal{L}$ and $X' \in \mathcal{N}$ we have $X\{X'\} \in \mathcal{N}$, $X[X'] \in \mathcal{N}$, and $X\langle X' \rangle \in \mathcal{N}$;
- for $X_1, \dots, X_n \in \mathcal{L}$ and $X'_1, \dots, X'_n \in \mathcal{N}$ we have $X_1(X'_1) \oplus \dots \oplus X_n(X'_n) \in \mathcal{N}$.

We call λ a *null attribute*, $X(X'_1, \dots, X'_n)$ a *record attribute*, $X\{X'\}$ a *set attribute*, $X[X']$ a *list attribute*, $X\langle X' \rangle$ a *multiset attribute* and $X_1(X'_1) \oplus \dots \oplus X_n(X'_n)$ a *union attribute*. As record and set attributes have a unique leading label, say X, we often write simply X to denote the attribute.

We can now extend the association *dom* from simple to nested attributes, i.e. for each $X \in \mathcal{N}$ we will define a set of values $dom(X)$.

Definition 3. For each nested attribute $X \in \mathcal{N}$ we get a *domain* $dom(X)$ as follows:

- $dom(\lambda) = \{\top\}$;
- $dom(X(X'_1, \dots, X'_n)) = \{(v_1, \dots, v_n) \mid v_i \in dom(X'_i) \text{ for } i = 1, \dots, n\}$;
- $dom(X\{X'\}) = \{\{v_1, \dots, v_k\} \mid k \in \mathbb{N} \text{ and } v_i \in dom(X') \text{ for } i = 1, \dots, k\}$, i.e. each element in $dom(X\{X'\})$ is a finite set with (pairwise different) elements in $dom(X')$;
- $dom(X[X']) = \{[v_1, \dots, v_k] \mid k \in \mathbb{N} \text{ and } v_i \in dom(X') \text{ for } i = 1, \dots, k\}$, i.e. each element in $dom(X[X'])$ is a finite (ordered) list with (not necessarily different) elements in $dom(X')$;
- $dom(X\langle X' \rangle) = \{\langle v_1, \dots, v_k \rangle \mid k \in \mathbb{N} \text{ and } v_i \in dom(X') \text{ for } i = 1, \dots, k\}$, i.e. each element in $dom(X\langle X' \rangle)$ is a finite multiset with elements in $dom(X')$, or in other words each $v \in dom(X')$ has a *multiplicity* $m(v) \in \mathbb{N}$ in a value in $dom(X\langle X' \rangle)$;
- $dom(X_1(X'_1) \oplus \dots \oplus X_n(X'_n)) = \{(X_i : v_i) \mid v_i \in dom(X'_i) \text{ for } i = 1, \dots, n\}$.

Note that the relational model is covered, if only the tuple constructor is used. Thus, instead of a relation schema R we will now consider a nested attribute X, assuming that the universe $\mathcal{U}$ and the set of labels $\mathcal{L}$ are fixed. Instead of an R-relation r we will consider a finite set $r \subseteq dom(X)$. An element of r is called a *tuple* or *complex value*. The following example includes several constructors.

Example 4. The nested attribute *Concert* allows to define an instance that contains data of a (rock-)concert.

Concert(Band(Bname(BandName),Members{Musician(
Name(MusicianName),Role(Instrument(InstrumentName)⊕Vocal(Voice)))}),
Played(Songs[SongTitle]),Evaluation(Grade)).

Here BandName, MusicianName, InstrumentName, Voice, SongTitle and Grade are simple attributes, while Concert, Band, Bname,Members, Musician, Name, Role, Instrument, Vocal, Played and Evaluation are labels. An element of the domain of nested attribute *Concert* could be the following tuple:
(∅,{ (`Greg Howe`,(`Instrument:Guitar`)),
(`Victor Wooten`,(`Instrument:Bassguitar`)),
(`Dennis Chambers`,(`Instrument:Drums`)),}
[`Tease, Contigo, Proto Cosmos`], `10`).
Note that this trio of jazz musicians plays under no specific band name.

In the following, we will need a bit more caution regarding syntax in order to avoid ambiguity. For this we mark the set label in an attribute of the form $X\{X_1(X_1') \oplus \cdots \oplus X_n(X_n')\}$ to indicate the inner union attribute, i.e. we should use $X_{\{1,\dots,n\}}$ (or even $X_{\{X_1,\dots,X_n\}}$) instead of X. As long as we are not dealing with subattributes of the form $X_{\{1,\dots,k\}}\{\lambda\}$, the additional index does not add any information and thus can be omitted to increase readability. The same applies to the multiset- and the list-constructor. The reason for introducing these indices will become apparent after Definition 6.

2.2 Subattributes

In the dependency theory for the relational model we considered the powerset $\mathcal{P}(R)$ for a relation schema R. $\mathcal{P}(R)$ is a Boolean algebra with order $\subseteq$, intersection $\cap$, union $\cup$ and the difference $-$.

We will generalize these operations for nested attributes starting with a partial order $\geq$. However, this partial order will be defined on equivalence classes of attributes. We will identify nested attributes, if we can identify their domains.

In the relational model a functional dependency $X \to Y$ for $X, Y \subseteq R \subseteq \mathcal{U}$ is satisfied by an R-relation r iff any two tuples $t_1, t_2 \in r$ that coincide on all the attributes in X also coincide on the attributes in Y. Crucial to this definition is that we can project R-tuples to subsets of attributes.

Therefore, in order to define FDs on a nested attribute $X \in \mathcal{N}$ we need a notion of subattribute. For this we define a partial order $\geq$ on nested attributes in such a way that whenever $X \geq Y$ holds, we obtain a canonical projection $\pi_Y^X : dom(X) \to dom(Y)$. However, this partial order has to be defined on equivalence classes of attributes, as some domains may be identified.

Definition 5. $\equiv$ is the smallest *equivalence relation* on $\mathcal{N}$ satisfying the following properties:

- $\lambda \equiv X()$;
- $X(X'_1, \dots, X'_n) \equiv X(X'_1, \dots, X'_n, \lambda)$;
- $X(X'_1, \dots, X'_n) \equiv X(X'_{\sigma(1)}, \dots, X'_{\sigma(n)})$ for any permutation $\sigma \in \mathbf{S}_n$;
- $X_1(X'_1) \oplus \cdots \oplus X_n(X'_n) \equiv X_{\sigma(1)}(X'_{\sigma(1)}) \oplus \cdots \oplus X_{\sigma(n)}(X'_{\sigma(n)})$ for any permutation $\sigma \in \mathbf{S}_n$;
- $X(X'_1, \dots, X'_n) \equiv X(Y_1, \dots, Y_n)$ if $X'_i \equiv Y_i$ for all $i = 1, \dots, n$;
- $X_1(X'_1) \oplus \cdots \oplus X_n(X'_n) \equiv X_1(Y_1) \oplus \cdots \oplus X_n(Y_n)$ if $X'_i \equiv Y_i$ for all $i = 1, \dots, n$;
- $X\{X'\} \equiv X\{Y\}$ iff $X' \equiv Y$;
- $X[X'] \equiv X[Y]$ iff $X' \equiv Y$;
- $X\langle X' \rangle \equiv X\langle Y \rangle$ iff $X' \equiv Y$;
- $X(X'_1, \dots, Y_1(Y'_1) \oplus \cdots \oplus Y_m(Y'_m), \dots, X'_n) \equiv Y_1(X'_1, \dots, Y'_1, \dots, X'_n) \oplus \cdots \oplus Y_m(X'_1, \dots, Y'_m, \dots, X'_n)$;
- $X\{X_1(X'_1) \oplus \cdots \oplus X_n(X'_n)\} \equiv X(X_1\{X'_1\}, \dots, X_n\{X'_n\})$;
- $X\langle X_1(X'_1) \oplus \cdots \oplus X_n(X'_n)\rangle \equiv X(X_1\langle X'_1\rangle, \dots, X_n\langle X'_n\rangle)$.

Basically, the first four cases in this equivalence definition state that λ in record attributes can be added or removed, and that order in record and union attributes does not matter. The last three cases in Definition 5 cover restructuring rules, two of which were already introduced in [2]. Obviously, if we have a set of labeled elements with up to n different labels, we can split this set into n subsets, each of which contains just the elements with a particular label, and the union of these sets is the original set. The same holds for multisets. Of course, we can also split a list of labeled elements into lists containing only elements with the same label, thereby preserving the order, but in this case we cannot invert the splitting and thus cannot claim an equivalence.

In the following we identify $\mathcal{N}$ with the set $\mathcal{N}/_{\equiv}$ of equivalence classes. In particular, we will write $=$ instead of $\equiv$, and in the following definition we should say that Y is a subattribute of X iff $\tilde{X} \geq \tilde{Y}$ holds for some $\tilde{X} \equiv X$ and $\tilde{Y} \equiv Y$.

Definition 6. For $X, Y \in \mathcal{N}$ we say that Y is a *subattribute* of X, iff $X \geq Y$ holds, where $\geq$ is the smallest partial order on $\mathcal{N}/_{\equiv}$ satisfying the following properties:

- $X \geq \lambda$ for all $X \in \mathcal{N}$;
- $X(Y_1, \dots, Y_n) \geq X(X'_{\sigma(1)}, \dots, X'_{\sigma(m)})$ for some injective $\sigma : \{1, \dots, m\} \to \{1, \dots, n\}$ and $Y_{\sigma(i)} \geq X'_{\sigma(i)}$ for all $i = 1, \dots, m$;
- $X_1(Y_1) \oplus \cdots \oplus X_n(Y_n) \geq X_{\sigma(1)}(X'_{\sigma(1)}) \oplus \cdots \oplus X_{\sigma(n)}(X'_{\sigma(n)})$ for some permutation $\sigma \in \mathbf{S}_n$ and $Y_i \geq X'_i$ for all $i = 1, \dots, n$;

- $X\{Y\} \geq X\{X'\}$ iff $Y \geq X'$;
- $X[Y] \geq X[X']$ iff $Y \geq X'$;
- $X\langle Y\rangle \geq X\langle X'\rangle$ iff $Y \geq X'$;
- $X_{\{1,\dots,n\}}[X_1(X_1') \oplus \cdots \oplus X_n(X_n')] \geq X(X_1[X_1'], \dots, X_n[X_n'])$;
- $X_{\{1,\dots,k\}}[X_1(X_1') \oplus \cdots \oplus X_k(X_k')] \geq X_{\{1,\dots,\ell\}}[X_1(X_1') \oplus \cdots \oplus X_\ell(X_\ell')]$ for $k \geq \ell$;
- $X(X_{i_1}\{\lambda\}, \dots, X_{i_k}\{\lambda\}) \geq X_{\{i_1,\dots,i_k\}}\{\lambda\}$;
- $X(X_{i_1}\langle\lambda\rangle, \dots, X_{i_k}\langle\lambda\rangle) \geq X_{\{i_1,\dots,i_k\}}\langle\lambda\rangle$;
- $X(X_{i_1}[\lambda], \dots, X_{i_k}[\lambda]) \geq X_{\{i_1,\dots,i_k\}}[\lambda]$.

Attributes of types $X_{\{i_1,\dots,i_k\}}\{\lambda\}$, $X_{\{i_1,\dots,i_k\}}\langle\lambda\rangle$ and $X_{\{i_1,\dots,i_k\}}[\lambda]$ are called *counter attributes*.

Note that the last four cases in Definition 6 cover further restructuring rules due to the union constructor. Obviously, if we are given a list of elements labeled with $X_1, \dots, X_n$, we can take the individual sublists – preserving the order – that contain only those elements labeled by X_i and build the tuple of these lists. In this case we can turn the label into a label for the whole sublist. This explains the first of the last four subattribute relationships.

For the other restructuring rules we have to add a little remark on notation here explaining why we use additional indices. As we identify $X\{X_1(X_1') \oplus \cdots \oplus X_n(X_n')\}$ with $X(X_1\{X_1'\}, \dots, X_n\{X_n'\})$, we obtain subattributes of the form $X(X_{i_1}\{X_{i_1}'\}, \dots, X_{i_k}\{X_{i_k}'\})$ for each subset $I = \{i_1, \dots, i_k\} \subseteq \{1, \dots, n\}$. However, restructuring requires some care with labels. If we simply reused the label X in the third last property in Definition 6, we would obtain

$$\begin{aligned} X\{X_1(X_1') \oplus X_2(X_2')\} \equiv & X(X_1\{X_1'\}, X_2\{X_2'\}) \geq \\ & X(X_1\{X_1'\}) \geq X(X_1\{\lambda\}) \geq X\{\lambda\}. \end{aligned}$$

However, the last step here is wrong, as the left hand side is an indicator for the subset containing the elements with label X_1 being empty or not, whereas the right hand side is the corresponding indicator for the whole set, i.e. elements with labels X_1 or X_2. No such mapping can be claimed. However, if we mark the set label in an attribute of the form $X\{X_1(X_1') \oplus \cdots \oplus X_n(X_n')\}$ to indicate the inner union attribute, the ambiguity problem disappears.

Further note that due to the restructuring rules in Definitions 5 and 6 we may have the case that a record attribute is a subattribute of a set attribute and vice versa. This cannot be the case, if the union-constructor is absent. However, the presence of the restructuring rules allows us to assume that the union-constructor only appears inside a set-constructor or as the outermost constructor. This will be frequently exploited in our proofs.

Obviously, $X \geq Y$ induces a projection map $\pi_Y^X : dom(X) \to dom(Y)$. For $X \equiv Y$ we have $X \geq Y$ and $Y \geq X$ and the projection maps π_Y^X and π_X^Y are inverse to each other.

Example 7. Let X = Balls{red(Number) $\oplus$ blue(Number) $\oplus$ green(Number)}. A complex value in $dom(X)$ represents a set of coloured balls carrying numbers, with the colours red, green and blue being available. Examples of such values are $v_1 = \{(\text{red} : 11), (\text{red} : 12), (\text{green} : 11), (\text{blue} : 6), (\text{blue} : 1)\}$, $v_2 = \{(\text{red} : 5), (\text{red} : 7), (\text{blue} : 3)\}$, and $v_3 = \{(\text{green} : 8)\}$.

Counter subattributes of X are $X_1 = \text{Balls}_{red,green}\{\lambda\}$, $X_2 = \text{Balls}_{green}\{\lambda\}$, and $X_3 = \text{Balls}_{blue}\{\lambda\}$. Projecting a value $v \in dom(X)$ to X_1 would give a non-empty set $\{\top\}$ iff v contains red or green balls. Analogously, the projection to X_2 or X_3 results in $\{\top\}$ iff v contains green or blue balls, respectively. For instance, we obtain

$$\begin{array}{lll} \pi^X_{X_1}(v_1) = \{\top\} & \pi^X_{X_2}(v_1) = \{\top\} & \pi^X_{X_3}(v_1) = \{\top\} \\ \pi^X_{X_1}(v_2) = \{\top\} & \pi^X_{X_2}(v_2) = \emptyset & \pi^X_{X_3}(v_2) = \{\top\} \\ \pi^X_{X_1}(v_3) = \{\top\} & \pi^X_{X_2}(v_3) = \{\top\} & \pi^X_{X_3}(v_3) = \emptyset \end{array}$$

We use the notation $\mathcal{S}(X) = \{Z \in \mathcal{N} \mid X \geq Z\}$ to denote the *set of subattributes* of a nested attribute X. Figure 2 shows the subattributes of $X\{X_1(A) \oplus X_2(B) \oplus X_3(C)\} = X(X_1\{A\}, X_2\{B\}, X_3\{C\})$ together with the relation $\geq$ on them.

Note that the subattribute $X\{\lambda\}$ would not occur, if we only considered the record-structure, whereas other subattributes such as $X(X_1\{\lambda\})$ would not occur, if we only considered the set-structure. This is a direct consequence of the restructuring rules.

Example 8. Consider the following subattributes of the nested attribute *Concert* of Example 4. Subattribute

Concert(Band(Members{Musician(Name(MusicianName))}))

represents the set of names of musicians performing at the concert. The projection of the tuple shown in Example 4 to this subattribute is the following complex value:

(({(`Greg Howe`),(`Victor Wooten`),(`Dennis Chambers`)})).

The subattribute Concert(Played(Songs[λ])) shows the number of songs played during the concert. The projection of the tuple of Example 4 to this subattribute is the tuple $(([\top, \top, \top]))$ showing that three songs were played. Finally, subattribute Concert(Band(Members{Musician(Role$_{\{\text{Vocal}\}}(\lambda)$)})) shows whether a singer performed at the concert. Projecting the tuple of Example 4 to this subattribute the tuple $((\{(\emptyset)\}))$ is obtained that shows that only instrumental music was played.

Let us now investigate the structure of $\mathcal{S}(X)$. We obtain a non-distributive lattice with relative pseudo-complements.

Definition 9. Let $\mathcal{L}$ be a lattice with zero and one, partial order $\leq$, join $\sqcup$ and meet $\sqcap$. $\mathcal{L}$ has *relative pseudo-complements* iff for all $Y, Z \in \mathcal{L}$ the infimum $Y \leftarrow Z = \bigsqcap\{U \mid U \sqcup Y \geq Z\}$ exists. Then $Y \leftarrow 1$ (1 being the one in $\mathcal{L}$) is called the *relative complement* of Y.

If we have distributivity in addition, we call $\mathcal{L}$ a *Brouwer algebra*. In this case the relative pseudo-complements satisfy $U \geq (Y \leftarrow Z)$ iff $(U \sqcup Y \geq Z)$, but if we do not have distributivity this property may be violated though relative pseudo-complements exist.

Proposition 10. *The set $\mathcal{S}(X)$ of subattributes carries the structure of a lattice with zero and one and relative pseudo-complements, where the order $\geq$ is as defined in Definition 6, and λ and X are the zero and one, respectively.*

It is easy to determine explicit inductive definitions of the operations $\sqcap$ (meet), $\sqcup$ (join) and $\leftarrow$ (relative pseudo-complement). This can be done by boring technical verification of the properties of meets, joins and relative pseudo-complements and is therefore omitted here.

Example 11. Let $X = X\{X_1(A) \oplus X_2(B)\}$ with $\mathcal{S}(X)$, as shown in Figure 1. Furthermore let $Y_1 = X_{\{1,2\}}\{\lambda\}$, $Y_2 = X(X_2\{B\})$, and $Z = X(X_1\{A\})$. Note that $\sqcup$ is the least common upper bound, while $\sqcap$ is the largest common lower bound in the subattribute poset. Then we have

$$\begin{gathered} Z \sqcap (Y_1 \sqcup Y_2) = X(X_1\{A\}) \sqcap (X_{\{1,2\}}\{\lambda\} \sqcup X(X_2\{B\})) = \\ X(X_1\{A\}) \sqcap X(X_1\{\lambda\}, X_2\{B\}) = X(X_1\{\lambda\}) \neq \lambda = \lambda \sqcup \lambda = \\ (X(X_1\{A\}) \sqcap X\{\lambda\}) \sqcup (X(X_1\{A\}) \sqcap X(X_2\{B\})) = (Z \sqcap Y_1) \sqcup (Z \sqcap Y_2) \ . \end{gathered}$$

This shows that $\mathcal{S}(X)$ in general is not a distributive lattice. Furthermore, $Y' \sqcup Z \geq Y_1$ holds for all Y' except λ, $X(X_1\{\lambda\})$ and $X(X_1\{A\})$. So $Z \leftarrow Y_1 = \lambda$, but not all $Y' \geq \lambda$ satisfy $Y' \sqcup Z \geq Y_1$.

2.3 Functional Dependencies

Let us now define functional and weak functional dependencies on $\mathcal{S}(X)$ and derive some sound derivation rules. The first thought would be to consider single nested attributes, as in the RDM $\sqcup$ corresponds to the union $\cup$, and $\sqcap$ to the intersection $\cap$. However, if we treat functional dependencies in this way, we cannot obtain a generalization of the extension rule. Therefore, we have to consider sets of subattributes.

Definition 12. Let $X \in \mathcal{N}$. A *functional dependency* (FD) on $\mathcal{S}(X)$ is an expression $\mathcal{Y} \to \mathcal{Z}$ with $\mathcal{Y}, \mathcal{Z} \subseteq \mathcal{S}(X)$. A *weak functional dependency* (wFD) on $\mathcal{S}(X)$ is an expression $\{\!\!\{\mathcal{Y}_i \to \mathcal{Z}_i \mid i \in I\}\!\!\}$ with an index set I and $\mathcal{Y}_i, \mathcal{Z}_i \subseteq \mathcal{S}(X)$.

In the following we consider finite sets $r \subseteq dom(X)$, which we will call simply *instances* of X.

Definition 13. Let r be an instance of X. We say that r *satisfies the FD* $\mathcal{Y} \to \mathcal{Z}$ on $\mathcal{S}(X)$ (notation: $r \models \mathcal{Y} \to \mathcal{Z}$) iff for all $t_1, t_2 \in r$ with $\pi^X_Y(t_1) = \pi^X_Y(t_2)$ for all $Y \in \mathcal{Y}$ we also have $\pi^X_Z(t_1) = \pi^X_Z(t_2)$ for all $Z \in \mathcal{Z}$.

r *satisfies the wFD* $\{\!\!\{\mathcal{Y}_i \to \mathcal{Z}_i \mid i \in I\}\!\!\}$ on $\mathcal{S}(X)$ (notation: $r \models \{\!\!\{\mathcal{Y}_i \to \mathcal{Z}_i \mid i \in I\}\!\!\}$) iff for all $t_1, t_2 \in r$ there is some $i \in I$ with $\{t_1, t_2\} \models \mathcal{Y}_i \to \mathcal{Z}_i$.

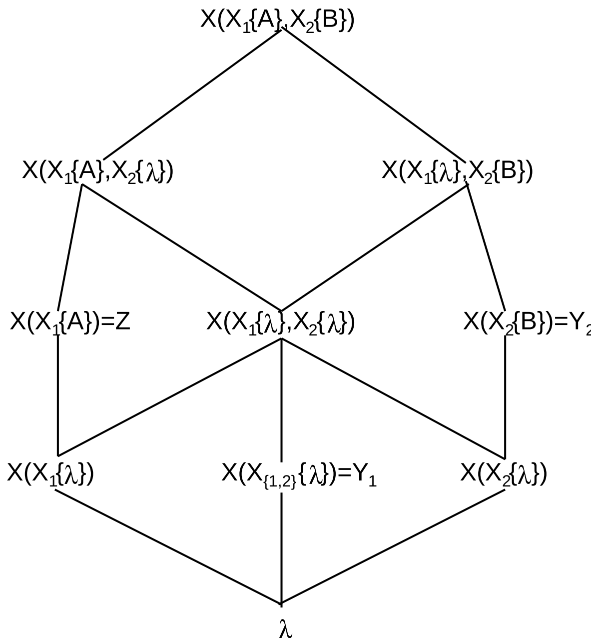

Figure 1: The lattice $\mathcal{S}(X\{X_1(A) \oplus X_2(B)\})$

According to this definition we identify a wfD $\{\!|\mathcal{Y} \to \mathcal{Z}|\!\}$, i.e. the index set contains exactly one element, with the "ordinary" FD $\mathcal{Y} \to \mathcal{Z}$.

2.4 Coincidence Ideals

The study of FDs and wFDs depends heavily on the notion of "coincidence ideal", i.e. sets of subattributes, on which two complex values coincide. For our purposes in this paper it is sufficient to take this as the definition.

In the following we investigate sets of subattributes, on which two complex values coincide. It is rather easy to see that these turn out to be ideals in the lattice $\mathcal{S}(X)$, i.e. they are non-empty and downward-closed. Therefore, we will call them *coincidence ideals*. However, there are many other properties that hold for coincidence ideals.

Definition 14. Two subattributes $Y, Z \in \mathcal{S}(X)$ are called *reconcilable* iff one of the following holds:

1. $Y \geq Z$ or $Z \geq Y$;
2. $X = X[X']$, $Y = X[Y']$, $Z = X[Z']$ and $Y', Z' \in \mathcal{S}(X')$ are reconcilable;
3. $X = X(X_1, \dots, X_n)$, $Y = X(Y_1, \dots, Y_n)$, $Z = X(Z_1, \dots, Z_n)$ and $Y_i, Z_i \in \mathcal{S}(X_i)$ are reconcilable for all $i = 1, \dots, n$;

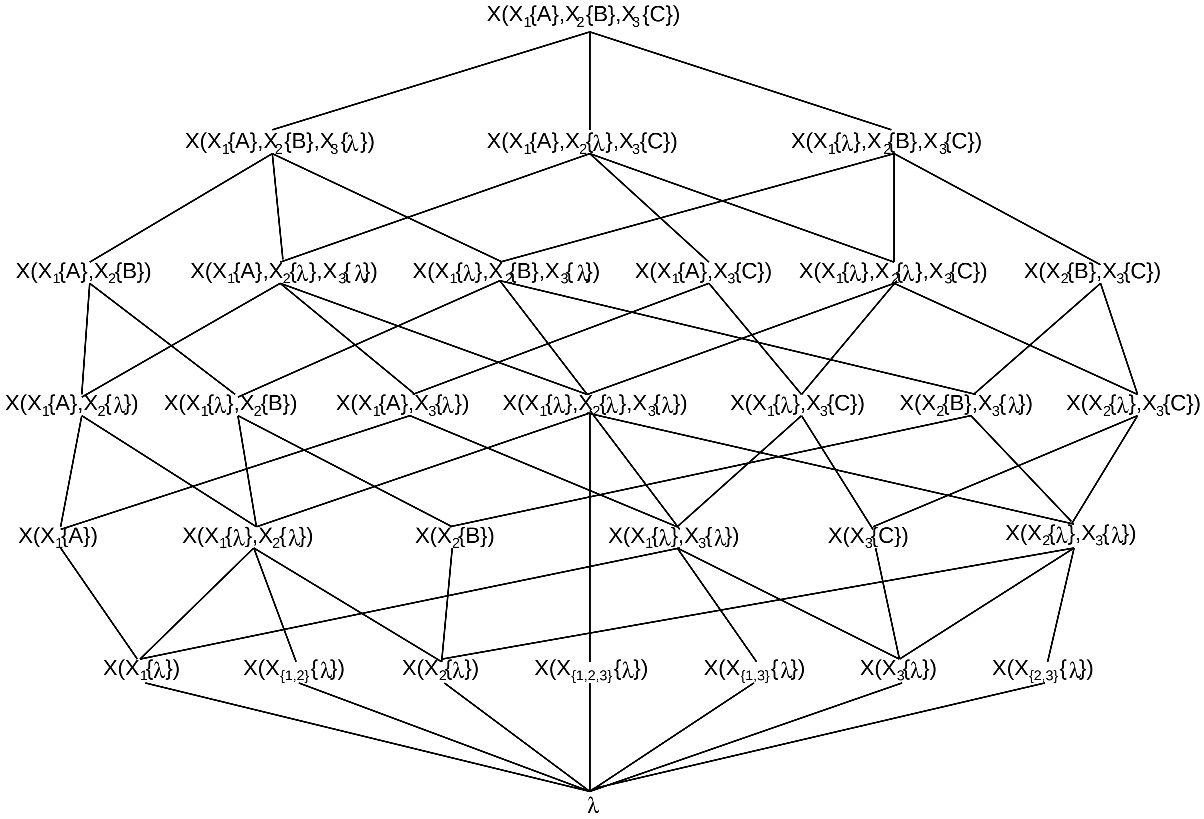

Figure 2: The lattice $\mathcal{S}(X\{X_1(A) \oplus X_2(B) \oplus X_3(C)\})$

4. $X = X_1(X'_1) \oplus \cdots \oplus X_n(X'_n)$, $Y = X_1(Y'_1) \oplus \cdots \oplus X_n(Y'_n)$, $Z = X_1(Z'_1) \oplus \cdots \oplus X_n(Z'_n)$ and $Y'_i, Z'_i \in \mathcal{S}(X'_i)$ are reconcilable for all $i = 1, \ldots, n$;

5. $X = X[X_1(X'_1) \oplus \cdots \oplus X_n(X'_n)]$, $Y = X(Y_1, \ldots, Y_n)$ with $Y_i = X_i[Y'_i]$ or $Y_i = \lambda = Y'_i$, $Z = X[X_1(Z'_1) \oplus \cdots \oplus X_n(Z'_n)]$, and Y'_i, Z'_i are reconcilable for all $i = 1, \ldots, n$.

Note that for the set- and multiset-constructor we can only obtain reconcilability for subattributes in a $\geq$-relation.

Example 15. Consider $\mathcal{S}(X\{X_1(A) \oplus X_2(B) \oplus X_3(C)\})$ shown on Figure 2. Here the subattributes $X(X_1\{A\}, X_2\{B\})$ and $X(X_1\{\lambda\}, X_2\{\lambda\}, X_3\{C\}$ are reconcilable. Indeed, $X(X_1\{A\}, X_2\{B\}) \equiv X(X_1\{A\}, X_2\{B\}, \lambda)$ and $X_1\{A\} \geq X_1\{\lambda\}$, $X_2\{B\} \geq X_2\{\lambda\}$, $\lambda \leq X_3\{C\}$, thus the components of the two subattributes are pairwise comparable in the subattribute lattice $\mathcal{S}(X)$, thus they are reconcilable by 1. of Definition 14. Applying 3. of the same definition the reconciliabiliy of $X(X_1\{A\}, X_2\{B\})$ and $X(X_1\{\lambda\}, X_2\{\lambda\}, X_3\{C\}$ is obtained.

On the other hand, $X(X_1\{A\}, X_2\{B\})$ and $X(X_{\{1,3\}}\{\lambda\})$ are not reconcilable.

The following definition of coincidence ideals looks formally self-referential. However, it is not hard to see that a *rank* of a nested attribute can be defined via the recursive construction as follows. The rank of a simple attribute is 0. When a nested attribute is constructed using some constructor, like record, set, list, multiset or disjoint union, then the rank of the new attribute is one plus the maximum rank of the parts it is constructed from. In this setting, whenever a coincidence ideal or defect coincidence ideal is referred to in the definition of coincidence ideal, then it is of subattributes of a nested attribute of strictly lower rank, hence there is no circularity in the definition.

Definition 16. A *coincidence ideal* on $\mathcal{S}(X)$ is a subset $\mathcal{F} \subseteq \mathcal{S}(X)$ with the following properties:

1. $\lambda \in \mathcal{F}$;

2. if $Y \in \mathcal{F}$ and $Z \in \mathcal{S}(X)$ with $Y \geq Z$, then $Z \in \mathcal{F}$;

3. if $Y, Z \in \mathcal{F}$ are reconcilable, then $Y \sqcup Z \in \mathcal{F}$;

4. a) if $X_I\{\lambda\} \in \mathcal{F}$ and $X_J\{\lambda\} \notin \mathcal{F}$ for $I \subsetneq J$, then $X(X_{i_1}\{X'_{i_1}\}, \ldots, X_{i_k}\{X'_{i_k}\}) \in \mathcal{F}$ for $I = \{i_1, \ldots, i_k\}$;

   b) if $X_I\{\lambda\} \in \mathcal{F}$ and $X(X_i\{\lambda\}) \notin \mathcal{F}$ for all $i \in I$, then there is a partition $I = I_1 \dot{\cup} I_2$ with $X_{I_1}\{\lambda\} \notin \mathcal{F}$, $X_{I_2}\{\lambda\} \notin \mathcal{F}$ and $X_{I'}\{\lambda\} \in \mathcal{F}$ for all $I' \subseteq I$ with $I' \cap I_1 \neq \emptyset \neq I' \cap I_2$;

   c) if $X_{\{1,\ldots,n\}}\{\lambda\} \in \mathcal{F}$ and $X_{I^-}\{\lambda\} \notin \mathcal{F}$ (for $I^- = \{i \in \{1, \ldots, n\} \mid X(X_i\{\lambda\}) \notin \mathcal{F}\}$), then there exists some $i \in I^+ = \{i \in \{1, \ldots, n\} \mid X(X_i\{\lambda\}) \in \mathcal{F}\}$ such that for all $J \subseteq I^-$ $X_{J \cup \{i\}}\{\lambda\} \in \mathcal{F}$ holds;

d) if $X_J\{\lambda\} \notin \mathcal{F}$ and $X_{\{j\}}\{\lambda\} \notin \mathcal{F}$ for all $j \in J$ and for all $i \in I$ there is some $J_i \subseteq J$ with $X_{J_i \cup \{i\}}\{\lambda\} \notin \mathcal{F}$, then $X_{I \cup J}\{\lambda\} \notin \mathcal{F}$, provided $I \cap J = \emptyset$;

e) if $X_{I^-}\{\lambda\} \in \mathcal{F}$ and $I' \subseteq I^+$ such that for all $i \in I'$ there is some $J \subseteq I^-$ with $X_{J \cup \{i\}}\{\lambda\} \notin \mathcal{F}$, then $X_{I' \cup J'}\{\lambda\} \notin \mathcal{F}$ for all $J' \subseteq I^-$ with $X_{J'}\{\lambda\} \notin \mathcal{F}$;

5. a) if $X_I\{\lambda\} \in \mathcal{F}$ and $X_J\{\lambda\} \in \mathcal{F}$ with $I \cap J = \emptyset$, then $X_{I \cup J}\{\lambda\} \in \mathcal{F}$;
   b) if $X_I[\lambda] \in \mathcal{F}$ and $X_J[\lambda] \in \mathcal{F}$ with $I \cap J = \emptyset$, then $X_{I \cup J}[\lambda] \in \mathcal{F}$;
   c) if $X_I\langle\lambda\rangle \in \mathcal{F}$ and $X_J\langle\lambda\rangle \in \mathcal{F}$ with $I \cap J = \emptyset$, then $X_{I \cup J}\langle\lambda\rangle \in \mathcal{F}$;
   d) if $X_I[\lambda] \in \mathcal{F}$ and $X_J[\lambda] \in \mathcal{F}$ with $J \subseteq I$, then $X_{I-J}[\lambda] \in \mathcal{F}$;
   e) if $X_I\langle\lambda\rangle \in \mathcal{F}$ and $X_J\langle\lambda\rangle \in \mathcal{F}$ with $J \subseteq I$, then $X_{I-J}\langle\lambda\rangle \in \mathcal{F}$;
   f) if $X_I[\lambda] \in \mathcal{F}$ and $X_J[\lambda] \in \mathcal{F}$, then $X_{I \cap J}[\lambda] \in \mathcal{F}$ iff $X_{(I-J)\cup(J-I)}[\lambda] \in \mathcal{F}$;
   g) if $X_I\langle\lambda\rangle \in \mathcal{F}$ and $X_J\langle\lambda\rangle \in \mathcal{F}$, then $X_{I \cap J}\langle\lambda\rangle \in \mathcal{F}$ iff $X_{(I-J)\cup(J-I)}\langle\lambda\rangle \in \mathcal{F}$;

6. a) for $X = X\{\bar{X}\{X_1(X_1') \oplus \cdots \oplus X_n(X_n')\}\}$, whenever $I \subseteq \{1, \dots, n\}$, there is a partition $I = I^- \cup I_{+-} \cup I_+ \cup I_-$ such that
      i. $X\{\bar{X}_{\{i\}}\{\lambda\}\} \in \mathcal{F}$ iff $i \notin I^-$,
      ii. $X\{\bar{X}_{I'}\{\lambda\}\} \in \mathcal{F}$, whenever $I' \cap I_+ \neq \emptyset$,
      iii. $X\{\bar{X}_{I'}\{\lambda\}\} \in \mathcal{F}$ iff $X\{\bar{X}_{I' \cap (I_{+-} \cup I^-)}\{\lambda\}\} \in \mathcal{F}$, whenever $I' \subseteq I_{+-} \cup I^- \cup I_-$;
   b) for $X = X\langle\bar{X}\{X_1(X_1') \oplus \cdots \oplus X_n(X_n')\}\rangle$, whenever $I \subseteq \{1, \dots, n\}$, there is a partition $I = I^- \cup I_{+-} \cup I_+ \cup I_-$ such that
      i. $X\langle\bar{X}_{\{i\}}\{\lambda\}\rangle \in \mathcal{F}$ iff $i \notin I^-$,
      ii. $X\langle\bar{X}_{I'}\{\lambda\}\rangle \in \mathcal{F}$, whenever $I' \cap I_+ \neq \emptyset$,
      iii. $X\langle\bar{X}_{I'}\{\lambda\}\rangle \in \mathcal{F}$ iff $X\langle\bar{X}_{I' \cap (I_{+-} \cup I^-)}\{\lambda\}\rangle \in \mathcal{F}$, whenever $I' \subseteq I_{+-} \cup I^- \cup I_-$;

7. a) if $X = X(X_1', \dots, X_n')$, then $\mathcal{F}_i = \{Y_i \in \mathcal{S}(X_i') \mid X(\lambda, \dots, Y_i, \dots, \lambda) \in \mathcal{F}\}$ is a coincidence ideal;
   b) if $X = X[X']$, such that X' is not a union attribute, and $\mathcal{F} \neq \{\lambda\}$, then $\mathcal{G} = \{Y \in \mathcal{S}(X') \mid X[Y] \in \mathcal{F}\}$ is a coincidence ideal;
   c) If $X = X_1(X_1') \oplus \cdots \oplus X_n(X_n')$ and $\mathcal{F} \neq \{\lambda\}$, then the set $\mathcal{F}_i = \{Y_i \in \mathcal{S}(X_i') \mid X_1(\lambda) \oplus \cdots \oplus X_i(Y_i) \oplus \cdots \oplus X_n(\lambda) \in \mathcal{F}\}$ is a coincidence ideal;
   d) if $X = X\{X'\}$, such that X' is not a union attribute, and $\mathcal{F} \neq \{\lambda\}$, then $\mathcal{G} = \{Y \in \mathcal{S}(X') \mid X\{Y\} \in \mathcal{F}\}$ is a defect coincidence ideal;
   e) if $X = X\langle X'\rangle$, such that X' is not a union attribute, and $\mathcal{F} \neq \langle\lambda\rangle$, then $\mathcal{G} = \{Y \in \mathcal{S}(X') \mid X\langle Y\rangle \in \mathcal{F}\}$ is a defect coincidence ideal.

A *defect coincidence ideal* on $\mathcal{S}(X)$ is a subset $\mathcal{F} \subseteq \mathcal{S}(X)$ satisfying properties 1, 2, 4(a)-(d), 6(a),(b), 7(d)-(e) and

8. a) if $X = X(X'_1, \dots, X'_n)$, then $\mathcal{F}_i = \{Y_i \in \mathcal{S}(X'_i) \mid X(\lambda, \dots, Y_i, \dots, \lambda) \in \mathcal{F}\}$ is a defect coincidence ideal;

   b) if $X = X[X']$, such that X' is not a union attribute, and $\mathcal{F} \neq \{\lambda\}$, then $\mathcal{G} = \{Y \in \mathcal{S}(X') \mid X[Y] \in \mathcal{F}\}$ is a defect coincidence ideal;

   c) If $X = X_1(X'_1) \oplus \cdots \oplus X_n(X'_n)$ and $\mathcal{F} \neq \{\lambda\}$, then the set $\mathcal{F}_i = \{Y_i \in \mathcal{S}(X'_i) \mid X_1(\lambda) \oplus \cdots \oplus X_i(Y_i) \oplus \cdots \oplus X_n(\lambda) \in \mathcal{F}\}$ is a defect coincidence ideal.

The name "coincidence ideal" was chosen, because these ideals characterize sets of subattributes, on which two complex values coincide. This is formally shown in the following theorem. In [16, 20] the term "SHL-ideal" was used instead; in [17] in a restricted setting the term "HL-ideal" was used. Note that in all these cases not all the conditions from Definition 16 were yet present.

For the purposes of the present paper the following three statements from [22] are important and not the particular details of the definition above.

Theorem 17 (Theorem 3.1 in [22])**.** *Let $X \in \mathcal{N}$ be a nested attribute. For complex values $t_1, t_2 \in dom(X)$ let $\mathcal{F} = \{Y \in \mathcal{S}(X) \mid \pi^X_Y(t_1) = \pi^X_Y(t_2)\} \subseteq \mathcal{S}(X)$ be the set of subattributes, on which they coincide. Then $\mathcal{F}$ is a coincidence ideal.*

Theorem 18 (Theorem 3.2 in [22])**.** *Let $\mathcal{G} \subseteq \mathcal{S}(X)$ be a defect coincidence ideal for the nested attribute $X \in \mathcal{N}$ such that the union constructor appears in X only directly inside a set-, list or multiset-constructor. Then the following holds:*

1. *There exist two finite sets $S_1, S_2 \subseteq dom(X)$ such that $\{\pi^X_Y(\tau) \mid \tau \in S_1\} = \{\pi^X_Y(\tau) \mid \tau \in S_2\}$ holds iff $Y \in \mathcal{G}$. For $\mathcal{G} \neq \{\lambda\}$ both sets are non-empty.*

2. *There exist two finite multisets $M_1, M_2 \subseteq dom(X)$ such that $\langle \pi^X_Y(\tau) \mid \tau \in M_1\rangle = \langle \pi^X_Y(\tau) \mid \tau \in M_2\rangle$ holds iff $Y \in \mathcal{G}$. For $\mathcal{G} \neq \{\lambda\}$ both multisets are non-empty.*

Theorem 19 (Central Theorem,Theorem 3.3 in [22])**.** *Let $\mathcal{F} \subseteq \mathcal{S}(X)$ be a coincidence ideal for the nested attribute $X \in \mathcal{N}$ such that the union constructor appears in X only directly inside a set-, list or multiset-constructor. Then there exist two complex values $t_1, t_2 \in dom(X)$ such that $\pi^X_Y(t_1) = \pi^X_Y(t_2)$ holds iff $Y \in \mathcal{F}$.*

The long and technical proofs of the above theorems are included in [22].

3 Keys and Antikeys

In this section we assume that a set Σ of functional dependencies is given over $\mathcal{S}(X)$ and every statement is understood as "with respect to Σ". Since functional dependencies are defined between *sets* of subattributes, the following is a natural generalization of the concept of keys to the higher-order datamodel.

Definition 20. $\mathcal{K} \subseteq \mathcal{S}(X)$ is a *key* (with respect to Σ) if $\Sigma \models \mathcal{K} \rightarrow \mathcal{S}(X)$ holds. In other words, if r is an instance of $\mathcal{S}(X)$ satisfying Σ, then for any two distinct complex valued tuples $t_1, t_2 \in r$ there exists $K \in \mathcal{K}$ such that $\pi_K^X(t_1) \neq \pi_K^X(t_2)$ holds.

The following closure operation is important in the characterization of minimal key systems.

Definition 21. The *closure* of a set $\mathcal{Y} \subseteq \mathcal{S}(X)$ is defined as the intersection of all coincidence-ideals containing $\mathcal{Y}$:

$$cl(\mathcal{Y}) = \bigcap_{\substack{\mathcal{F} \text{is a coincidence-ideal} \\ \mathcal{Y} \subseteq \mathcal{F}}} \mathcal{F}. \tag{1}$$

The idea behind Definition 21 is simple. We are interested in the following: assume that two tuples agree on a set of subattributes, where do they need to agree besides those? Since it was proved in [22] that the set of subattributes where two tuples coincide form a coincidence ideal, if $\pi_Y^X(t_1) = \pi_Y^X(t_2)$ for all $Y \in \mathcal{Y}$, then $\mathcal{Y}$ is a subset of the set of subattributes where t_1, t_2 coincide, which is a coincidence ideal. Because $cl(\mathcal{Y})$ is a subset of that ideal, t_1, t_2 coincide on all $Z \in cl(\mathcal{Y})$.

Proposition 22. *The operator cl is a closure operator, that is*

1. $\mathcal{Y} \subseteq cl(\mathcal{Y})$;
2. *If* $\mathcal{Y} \subseteq \mathcal{Z}$, *then* $cl(\mathcal{Y}) \subseteq cl(\mathcal{Z})$;
3. $cl(cl(\mathcal{Y})) = cl(\mathcal{Y})$.

Clearly, if $\mathcal{K}$ is a key and $\mathcal{K} \subset \mathcal{H}$, then $\mathcal{H}$ is a key, as well. In particular, the closure of a key is also a key. The interesting fact is that the converse also holds.

Theorem 23. $\mathcal{K} \subseteq \mathcal{S}(X)$ *is a key iff* $cl(\mathcal{K})$ *is a key.*

Proof. According to the previous note, only the implication "$cl(\mathcal{K})$ is a key $\Longrightarrow \mathcal{K}$ is a key" needs to be proven. Suppose that $\mathcal{K}$ is not a key, that is $\Sigma \not\models \mathcal{K} \rightarrow \mathcal{S}(X)$. Thus, there exists an instance r of $\mathcal{S}(X)$ satisfying Σ and two complex-valued tuples $t_1 \neq t_2$ in r such that $\forall K \in \mathcal{K}\colon \pi_K^X(t_1) = \pi_K^X(t_2)$ holds. Let $\mathcal{F} = \{Z \mid \pi_Z^X(t_1) = \pi_Z^X(t_2)\}$. Since $\mathcal{F}$ is a coincidence-ideal that contains $\mathcal{K}$, $cl(\mathcal{K}) \subseteq \mathcal{F}$ holds. This implies, that $cl(\mathcal{K})$ is not a key either.

□

Antikeys are defined in the relational model as any subset of attributes that are not keys. Here, the same works.

Definition 24. A subset $\mathcal{A} \subset \mathcal{S}(X)$ is an *antikey* (with respect to Σ), if $\Sigma \not\models \mathcal{A} \rightarrow \mathcal{S}(X)$. In other words, there exists an instance r of $\mathcal{S}(X)$ satisfying Σ and two complex-valued tuples $t_1 \neq t_2$ in r such that $\forall A \in \mathcal{A}\colon \pi_A^X(t_1) = \pi_A^X(t_2)$ holds.

It is clear, that if $\mathcal{A}$ is an antikey, and $\mathcal{B} \subseteq \mathcal{A}$, then $\mathcal{B}$ is an antikey, as well. In particular, if $cl(\mathcal{A})$ is an antikey, then so is $\mathcal{A}$, as well. Again, the interesting fact is that the converse is also true follows from Theorem 23.

Corollary 25. $\mathcal{A} \subset \mathcal{S}(X)$ *is an antikey, iff* $cl(\mathcal{A})$ *is an antikey.*

Theorem 23 and Corollary 25 allow considering only closed sets as keys or antikeys. $\mathcal{H}$ is *closed* if $\mathcal{H} = cl(\mathcal{H})$. Indeed, if we have a key $\mathcal{K}$, then its closure $cl(\mathcal{K})$ is also a key and every $\mathcal{K}'$ with $cl(\mathcal{K}') = cl(\mathcal{K})$ is a key. as well. This means that the system of closed sets that are keys uniquely determines the system of all keys. Thus, we concentrate on only closed sets in the following.

We are interested in *minimal keys* and *maximal antikeys*, where minimal and maximal is with respect (set-wise) containment. Given Σ, the system $\mathfrak{K} = \{\mathcal{K}_1, \ldots, \mathcal{K}_k\}$ of all minimal keys forms a *Sperner system* or *antichain* of sets of subattributes, that is for every pair of indices i and j $\mathcal{K}_i \subsetneqq \mathcal{K}_j$ holds. Analogously, maximal antikeys form a Sperner system $\mathfrak{A} = \{\mathcal{A}_1, \ldots, \mathcal{A}_a\}$ of sets of subattributes.

Proposition 26. *The system of minimal keys and the system of maximal antikeys mutually determine each other.*

Proof. Consider the poset $\mathfrak{P}$ of closed subsets of $\mathcal{S}(X)$ ordered by (set-wise) inclusion. Keys form an *up-set*, or *filter*, that is a subset $\mathbb{K} \subseteq \mathfrak{P}$ such that if $\mathcal{K} \in \mathbb{K}$ and $\mathcal{K} \subset \mathcal{H}$, then $\mathcal{H} \in \mathbb{K}$. Similarly, antikeys form a *down-set*, or *ideal*, that is a subset $\mathbb{A} \subseteq \mathfrak{P}$ such that if $\mathcal{A} \in \mathbb{A}$ and $\mathcal{B} \subset \mathcal{A}$, then $\mathcal{B} \in \mathbb{A}$. Most importantly, $\mathbb{K} \cup \mathbb{A} = \mathfrak{P}$ and $\mathbb{K} \cap \mathbb{A} = \emptyset$. The system of minimal keys is the set of minimal elements of $\mathbb{K}$, while the system of maximal antikeys is the set of maximal elements of $\mathbb{A}$.

□

Proposition 26 allows the following notation. If $\mathfrak{K}$ is the system of minimal keys, then the corresponding system of maximal antikeys $\mathfrak{A}$ is denoted by $\mathfrak{K}^{-1}$, as well. Observe, that this notation can be extended to any Sperner-system $\mathfrak{S}$ of closed sets by $\mathfrak{S}^{-1}$ being the collection of the maximal elements of the ideal that is the complement in $\mathfrak{P}$ of the filter $\mathbb{S}$ generated by $\mathfrak{S}$.

4 Armstrong Instances

The principal interest of the present paper is to investigate which Sperner systems of closed subsets of $\mathcal{S}(X)$ can occur as systems of minimal keys for some suitable family of functional dependencies Σ. The idea of Armstrong instance is that given a family of constraints (e.g. functional dependencies) and a subset Σ of that family, one looks for a model (database) that satisfies only those constraints in Σ and no others. The practical use of this concept is that during conceptual schema design, the designer is able to check whether some constraints are logical consequences of the constraints of the design by obtaining an Armstrong instance and checking what dependencies are satisfied besides the ones designed. For the relational model, there are even software packages constructing Armstrong instances.

In the relational model Armstrong [4] and Demetrovics [6] proved that *every* Sperner system arises as set of minimal keys, i.e., has an Armstrong instance. Later Fagin [10] gave necessary and sufficient conditions for constraints that can be described by Horn clauses, to have Armstrong instance in the framework of the relational datamodel. However, in [21] it was shown that in the higher-order datamodel, although in the restricted "counter-free" case, the same statement does not hold.

Definition 27. Let r be an instance of a nested attribute X, with subattribute lattice $\mathcal{S}(X)$. A subset $\mathcal{K} \subseteq \mathcal{S}(X)$ is *key with respect to* r, if $r \models \mathcal{K} \rightarrow \mathcal{S}(X)$, i.e., there exist no two distinct complex-valued tuples $t_1, t_2 \in r$ such that $\forall K \in \mathcal{K}\colon \pi_K^X(t_1) = \pi_K^X(t_2)$ holds. r is an *Armstrong-instance* for a Sperner system $\mathfrak{K}$ of closed subsets of $\mathcal{S}(X)$, if the system of minimal keys with respect to r is exactly $\mathfrak{K}$.

A simple characterization can be given for Armstrong instances.

Proposition 28. *Let $\mathfrak{K}$ be a Sperner system of closed subsets of $\mathcal{S}(X)$. An instance r is Armstrong-instance for minimal key system $\mathfrak{K}$ iff*

[Key] *For all $\mathcal{K} \in \mathfrak{K}$ and any two complex-valued tuples $t_1, t_2 \in r$ there exists $K \in \mathcal{K}$ such that $\pi_K^X(t_1) \neq \pi_K^X(t_2)$ holds.*

[Antikey] *For all $\mathcal{A} \in \mathfrak{K}^{-1}$ there exists two complex-valued tuples $t_1 \neq t_2$ in r such that $\forall A \in \mathcal{A}\colon \pi_A^X(t_1) = \pi_A^X(t_2)$ holds.*

□

The [**Antikey**] property of Proposition 28 gives an immediate necessary condition for existence of an Armstrong-instance. Indeed, every maximal antikey must be a coincidence ideal. This is a real restriction, since not all closed sets are coincidence ideals. For example, consider $\mathcal{S}(X\{X_1(A) \oplus X_2(B) \oplus X_3(C)\})$ of Figure 2. For the sake of convenience the principal ideal $\{Y \in \mathcal{S}(X) \mid Y \leq Z\}$ of $\mathcal{S}(X)$ generated by $Z \in \mathcal{S}(X)$ is denoted by $Z{\downarrow}$. The principal ideal $\mathcal{I} = X(X_1\{\lambda\}, X_2\{\lambda\}){\downarrow}$ is closed see Proposition 37, but not a coincidence ideal, since it violates property 4(a) of Definition 16. Taking $\mathbb{K} \subset \mathfrak{P}$ be the set of closed subsets of $\mathcal{S}(X)$ that do not contain $\mathcal{I}$, we obtain that the unique maximal antikey corresponding to the key system $\mathbb{K}$ is $\mathcal{I}$. However, since $\mathcal{I}$ is not a coincidence ideal, $\mathbb{K}$ cannot have an Armstrong-instance.

On the other hand, minimal keys can indeed be closed sets that are not coincidence ideals. Consider again $\mathcal{S}(X\{X_1(A) \oplus X_2(B) \oplus X_3(C)\})$ of Figure 2. Let $\mathcal{A} = X(X_1\{A\}){\downarrow}$. It is not hard to see that $\mathcal{A}$ is a coincidence ideal. Indeed, properties 1-3 of Definition 16 are trivially satisfied by any principal ideal. Property 4(a) is satisfied, because the only possible choice of I that satisfies the conditions is $I = \{1\}$. Conditions in points (b), (c), and (e) of property 4 do not apply to $\mathcal{A}$, while 4(d) is satisfied trivially. Finally, the conditions in properties 5-7 do not apply to $\mathcal{A}$, hence by the Central Theorem (Theorem 19) there exists two tuples

$t_1, t_2 \in dom(X)$ with $\pi_A^X(t_1) = \pi_A^X(t_2)$ iff $A \in \mathcal{A}$. In fact the proof of the Central Lemma constructs the tuples $t_1 = \emptyset$ and $t_2 = \{(X_1 : a_1)\}$. $\mathcal{K} = X(X_1\{\lambda\}, X_2\{\lambda\})\downarrow$ is a minimal key with respect to the instance $r = \{t_1, t_2\}$.

The trouble with Armstrong-instances are caused by *degenerate keys*.

Definition 29. A key $\mathcal{K}$ is called *degenerate*, if every $K \in \mathcal{K}$ is constructed using only λ, set-constructor, record-constructor and union-constructor. That is, K does not contain simple attributes, multiset- or list-constructors.

Similar question was considered by Fagin and Vardi for the relational model in [11], where functional dependencies with non-empty left hand side were called *standard*, and the problems of working with non-standard functional dependencies were investigated. The following theorem gives a sufficient condition for the existence of Armstrong-instance.

Theorem 30. *Let $\mathfrak{K} = \{\mathcal{K}_i \mid i = 1, 2, \ldots, k\}$ be a Sperner system of closed subsets of $\mathcal{S}(X)$. There exists an Armstrong-instance r for $\mathfrak{K}$ as system of minimal keys provided the following two conditions hold:*

1. *$\mathfrak{K}$ does not contain degenerate keys;*
2. *Each element of $\mathfrak{A} = \mathfrak{K}^{-1}$ is a coincidence ideal.*

Proof. Let $\mathfrak{K}^{-1} = \{\mathcal{A}_1, \ldots, \mathcal{A}_k\}$. The restructuring rules allow us to assume that the union-constructor only appears inside a set-constructor or as the outermost constructor, hence Theorem 19 provides complex values t_0^i, t_1^i $i = 1, 2, \ldots, k$ such that

$$\pi_Y^X(t_0^i) = \pi_Y^X(t_1^i) \iff Y \in \mathcal{A}_i. \tag{2}$$

This ensures that $\mathcal{A}_i$ is an antikey for all i. On the other hand, we have to show that each $\mathcal{K}_i \in \mathfrak{K}$ is a key. In order to do so, the complex valued tuples will be modified preserving (2) so that if $\pi_Z^X(t_a^i) = \pi_Z^X(t_b^j)$ for some $Z \in \mathcal{S}(X)$, $a, b \in \{0, 1\}$, and $1 \le i < j \le k$, then Z cannot contain simple attributes or list or multiset subattributes. Hence no two complex values can agree on every subattribute in $\mathcal{K}_s$ for all s, which implies that each $\mathcal{K}_s$ is a key. Note, that the number of complex values in this Armstrong instance is exactly $2|\mathfrak{K}^{-1}|$.

The modification of the tuples is as follows. For simple attributes we have to take care of that during the inductive construction of the tuples the constants from the domains of simple attributes used for $\mathcal{A}_i$ must be distinct from those used for $\mathcal{A}_j$ if $i \neq j$. This ensures that values constructed for $\mathcal{A}_i$ and those constructed for $\mathcal{A}_j$ for $i \neq j$ cannot agree on subattributes containing a simple attribute.

For the list attribute case if one of $\mathcal{A}_i$'s is $\{\lambda\}$, then we have only one coincidence ideal by the Sperner property, so there is nothing to prove. Otherwise, consider the inductive construction of the tuples t_0^i, t_1^i for $\mathcal{A}_i$. We modify that in a sequential order for $i = 2, 3, \ldots, k$. When we encounter a list subattribute $X[X']$, (X' could be a union) the proof of Theorem 19 constructs two tuples $t_0^{i\prime}, t_1^{i\prime}$ that are of the form $t_a^{i\prime} = [t_a^{i\prime\prime}]$, $a = 0, 1$. Let m be the largest multiplicity of any element in any list in t_0^j and t_1^j, $j = 1, 2, \ldots, i-1$. Now, we replace $t_0^{i\prime}, t_1^{i\prime}$ with $t_a^{i*} = [(m+1) \cdot t_a^{i\prime\prime}]$, i.e. t_a^{i*} is a list with $m+1$ occurrences of the same element $t_a^{i\prime\prime}$. This modification ensures that multiplicities inside lists cannot agree in tuples constructed for distinct $\mathcal{A}_i$'s while preserving the property (2). The multiset attribute case is similar.

□

4.1 The Case $\mathcal{S}(X\{X_1(A_1) \oplus X_2(A_2) \oplus \cdots \oplus X_n(A_n)\})$

In the present section we study a special case, which is archetypical. This nested attribute exhibits most of the problems with respect to Armstrong-instances, thus showing the complexity of the problem. We believe that effective treatment of this case would lead to general insight of the nature of Armstrong-instances of nested attributes. As a beginning in that direction, a characterization is given for the existence of such instances. Let r be an instance of $\mathcal{S}(X\{X_1(A_1) \oplus X_2(A_2) \oplus \cdots \oplus X_n(A_n)\})$. According to Definition 5, complex value $t \in r$ can be considered as a tuple $t = (X_1 : a_1, \ldots, X_n : a_n)$, where a_i is a finite subset of the domain of A_i for $i = 1, 2, \ldots, n$. The *pattern of* t is an n-tuple p_t of $+$'s and $-$'s, such that the i^{th} coordinate of p_t is $+$, if $a_i \neq \emptyset$, and $-$, if $a_i = \emptyset$.

Proposition 31. *Let r be an instance of $X\{X_1(A_1) \oplus X_2(A_2) \oplus \cdots \oplus X_n(A_n)\}$, and let $\mathfrak{K} = \{\mathcal{K}_1, \ldots, \mathcal{K}_k\}$ be the system of minimal keys with respect to r. If there exists an i such that $\mathcal{K}_i$ is degenerate, then r contains at most one complex valued tuple of each possible pattern. Consequently, $|r| \leq 2^n$.*

Proof. Attributes in a degenerate key can only have the form $X_I\{\lambda\}$ for some $I \subseteq \{1, 2, \ldots, n\}$ or $X(X_{i_1}\{\lambda\}, X_{i_2}\{\lambda\}, \ldots X_{i_s}\{\lambda\})$. The projection of a complex-valued tuple t to such an attribute is determined by which coordinates of t are non-empty, hence depend only on the patter p_t.

□

Any two subattributes that are not of type $X_I\{\lambda\}$ for some $|I| > 1$ are reconcilable since the possible i^{th} components of a tuple are $\lambda, X_i\{\lambda\}$ and $X_i\{A_i\}$ that are pairwise comparable, that is reconcilable. Thus, a coincidence ideal $\mathcal{A}$ contains the $\sqcup$ of any pair of non-counter attributes belonging to $\mathcal{A}$. It follows then that $\mathcal{A}$ consists of a principal ideal of non-counter attributes extended with some counter attributes of type $X_I\{\lambda\}$. Recall, that a *principal ideal* generated by an element κ in a lattice consists of all elements μ of that lattice with $\mu \leq \kappa$.

Take a Sperner system of closed sets $\mathfrak{K} = \{\mathcal{K}_1, \ldots, \mathcal{K}_k\}$ and the Sperner system $\mathfrak{K}^{-1} = \mathfrak{A} = \{\mathcal{A}_1, \ldots, \mathcal{A}_m\}$ as candidates for minimal keys and maximal antikeys, respectively. Assuming that each $\mathcal{A}_i$ is a coincidence ideal, a pair of tuple patterns is obtained via Theorem 19 for each $\mathcal{A}_i$, together with a constraint $\varphi_{i_1} \wedge \ldots \wedge \varphi_{i_b}$, where φ_{i_j} requires $=$ or $\neq$ on common $+$-component i_j. Indeed, let $X(Y_1, Y_2, \ldots, Y_n)$ be the largest (generator) element of the principal ideal part of $\mathcal{A}_i$. If both tuple patterns contain $+$ in component i_j, then either $Y_{i_j} = X_{i_j}\{\lambda\}$ or $Y_{i_j} = X_{i_j}\{A_{i_j}\}$. In the first case the tuples that agree on exactly $\mathcal{A}_i$ must contain different nonempty sets in the i_jth component giving φ_{i_j} being $\neq$. Note that the tuples cannot agree on $X_{i_j}\{A_{i_j}\}$ in this case. On the other hand, if $Y_{i_j} = X_{i_j}\{A_{i_j}\}$, then the i_jth component of the tuples must contain the same non-empty set, thus φ_{i_j} is $=$.

Proposition 32. *The pair of patterns and the constraints are uniquely determined by the counter attributes contained in $\mathcal{A} \in \mathfrak{A}$, provided the pair consists of distinct patterns.*

Proof. Assume that $\mathcal{A} = X(Y_1, Y_2, \ldots, Y_n)\downarrow\cup\{X_I \mid I \in \mathcal{I}\}$ where Y_i is either λ, $X_i\{\lambda\}$ or $X_i\{A_i\}$. Furthermore, assume that two complex valued tuples $t_1 = (X_1 : a_1, \ldots, X_n : a_n)$ and $t_2 = (X_1 : b_1, \ldots, X_n : b_n)$ agree exactly on subattributes of $\mathcal{A}$. It means that if $Y_i = \lambda$, then one of a_i and b_i is empty and the other is nonempty, if $Y_i = X_i\{\lambda\}$, then $a_i \neq b_i$ and both are nonempty, while if $Y_i = X_i\{A_i\}$, then either $a_i = b_i$ and both are nonempty, or both are empty. Thus, patterns of t_1 and t_2 have the same symbol in coordinate j where $Y_j = X_i\{A_j\}$ or $Y_j = X_j\{\lambda\}$, and opposite symbols for $Y_j = \lambda$. $X_I\{\lambda\} \in \mathcal{A}$ for $|I| > 1$ means that both t_1 and t_2 have a nonempty coordinate whose index is in I, where the nonempty coordinates of t_1 and t_2 showing $X_I\{\lambda\} \in \mathcal{A}$ need not have the same index. Let us assume that $Y_{i_0} = \lambda$ and t_1 has empty $i_0{}^{\text{th}}$ coordinate. If $X_{\{i_0,j\}}\{\lambda\} \in \mathcal{A}$, then the j^{th} coordinate of t_1 is nonempty. On the other hand, if $X_{\{i_0,j\}}\{\lambda\} \notin \mathcal{A}$, then j^{th} coordinate of t_1 is empty. In both cases the j^{th} coordinate of pattern of t_2 is uniquely determined, as well.

The remaining case is when none of Y_i's is λ. In this case, using the same argument as before, the patterns of t_1 and t_2 are the same.

□

For example, if $\mathcal{A}_i = X(X_1\{A_1\}, X_2\{\lambda\})\downarrow\cup\{X_I \mid I \cap \{1,2\} \neq \emptyset\}$, then the values $\{(X_1 : v_1), (X_2 : v_2)\}$ and $\{(X_1 : v_1), (X_2 : v_2'), (X_3 : v_3), \ldots, (X_n : v_n)\}$ coincide exactly on $\mathcal{A}_i$.

The corresponding pair of tuple patterns is $\{(+,+,-\ldots,-), (+,+,\ldots,+)\}$, and the constraints are $\varphi_1\colon =_1$, $\varphi_2\colon \neq_2$.

Construct a graph on 2^n vertices with vertex set V being all possible patterns. Add a *green* edge between the two patterns given by $\mathcal{A}_i$ labeled with the appropriate constraint, for each candidate antikey $\mathcal{A}_i$. For each pair of patterns $\mathfrak{K}$ defines (a more complicated) constraint on the patterns. That is, each $\mathcal{K}_j$ need to have an element K where the two tuples corresponding to the pair of patterns have distinct projections. For each K, a disjunction of conjuncts can be formulated, and the disjunction Φ_j of these formulae expresses that $\mathcal{K}_j$ is a key. Finally, the constraint on the pair of patterns defined by $\mathfrak{K}$ is $\Phi_1 \wedge \ldots \wedge \Phi_k$. Add *red* edge between two patterns labeled by the appropriate constraint.

Theorem 33. *Let $\mathfrak{K} = \{\mathcal{K}_1, \ldots, \mathcal{K}_k\}$ be a Sperner system of closed sets that contains a degenerate key and assume that $\mathfrak{K}^{-1} = \mathfrak{A} = \{\mathcal{A}_1, \ldots, \mathcal{A}_m\}$ consists of coincidence ideals. Let G be the graph on the patterns with green and red labeled edges as constructed above. $\mathfrak{K}$ has an Armstrong-instance iff the subgraph G' spanned by the* green *edges has edge labels (both green and red), that can be simultaneously satisfied by a set r of tuples that contain tuples of each pattern of G'.*

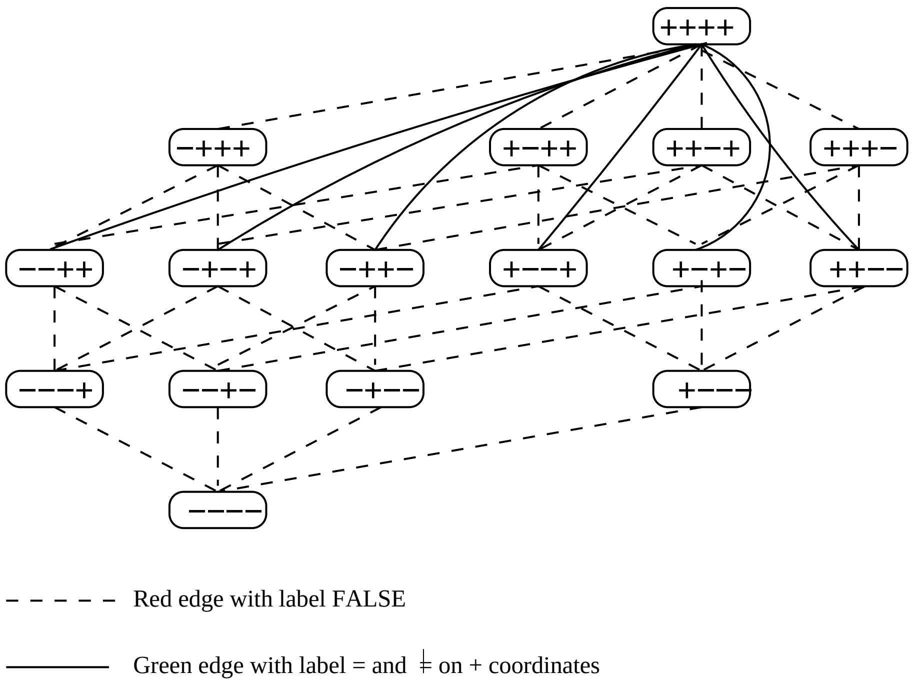

Figure 3: Pattern graph

Proof. If $\mathfrak{K} = \{\mathcal{K}_1, \ldots, \mathcal{K}_k\}$ has an Armstrong-instance r, then the patterns of tuples in r satisfy each edge constraint. On the other hand, assume that the edge labels of the subgraph G' spanned by the green edges are simultaneously satisfiable, let r be a set of tuples that satisfy all constraints in G'. Since there is a degenerate candidate key in $\mathfrak{K}$, $X(X_1\{\lambda\}, X_2\{\lambda\}, \ldots, X_n\{\lambda\})$ cannot be contained in any of the candidate antikeys, hence by Proposition 32 there is a unique green edge with label for each $\mathcal{A}_i$. The pair of tuples corresponding to the endpoints of the edge agree exactly on subattributes of $\mathcal{A}_i$, showing that it is an antikey. The red edge labels make sure that each $\mathcal{K}_i$ is a key. Since $\mathfrak{K}^{-1} = \mathfrak{A}$, we have that $\mathfrak{K}$ is the system of minimal keys, $\mathfrak{A}$ is the system of maximal antikeys with respect to r.

□

Example 34. Let $n = 4$. For all four choices of $1 \leq i < j < k \leq 4$ let $\mathfrak{K}$ consist of the principal ideals $X(X_i\{\lambda\}, X_j\{\lambda\}, X_k\{\lambda\})\downarrow$. Then $\mathfrak{K}^{-1} = \mathfrak{A}$ consist of $\mathcal{A}_{i,j} = X(X_i\{A_i\}, X_j\{\lambda\})\downarrow \cup \{X_I \mid I \cap \{i,j\} \neq \emptyset\}$ for all six choices of $1 \leq i < j \leq 4$. The pattern graph for $\mathfrak{K}$ and $\mathfrak{A}$ is shown on Figure 3.

For instance, between $+ + ++$ and $- - ++$ we have a green edge with label $\varphi_3 \wedge \varphi_4$ given by $\varphi_3 :=_3$ and $\varphi_4 :\neq_4$. This edge originates from $\mathcal{A}_{3,4}$. Between $- + ++$ and $- - ++$ we get a red edge with label FALSE, because the key $\mathcal{K} =$

$X(X_1\{\lambda\}, X_2\{\lambda\}, X_3\{\lambda\})\downarrow$ with $K = X(X_1\{\lambda\}, X_2\{\lambda\}, X_3\{\lambda\})$ will always yield inequality for the second component. Similarly, between $---+$ and $--+-$ we get a red edge with label TRUE, because each key $\mathcal{K}_{ijk} = X(X_i\{\lambda\}, X_j\{\lambda\}, X_k\{\lambda\})\downarrow$ contains $K = X(X_3\{\lambda\})$ or $K = X(X_4\{\lambda\})$, so we know the required inequality will be satisfied.

The red (dotted) edges with constraint label TRUE are not drawn. They are between pairs of vertices that have at least two coordinates where one of them is + and the other is -, that is exactly the complement of the drawn dotted graph. It is easy to see that the labels on the subgraph induced by the green (continuous line) edges are satisfiable, $r = \{(X_1 : a, X_2 : b, X_3 : c, X_4 : d), (X_1 : a, X_2 : b'), (X_1 : a, X_3 : c'), (X_1 : a, X_4 : d'), (X_2 : b, X_3 : c'), (X_2 : b, X_4 : d'), (X_3 : c, X_4 : d')\}$ is an Armstrong-instance.

Note that the Armstrong instance constructed in Example 34 contains a value for each of the edges in the subgraph of the pattern graph spanned by the green edges. This is the construction used in the proof of Theorem 33.

4.1.1 Some Negative Results.

In [21] some examples were shown that did not have Armstrong-instance in the counter-free case. The proofs there were sometimes quite involved, which was caused by not considering the counter attributes. If those are taken into account, the proofs can be shortened, since the counter attributes contained in the maximal antikeys sort of determine the patterns of possible complex values in an Armstrong-instance.

Example 35. This example is from [21], but the proof is much shorter. Let $X = X\{X_1(A_1) \oplus X_2(A_2) \oplus X_3(A_3) \oplus X_4(A_4)\}$ and consider $\mathcal{S}(X)$. Let the Sperner system $\mathfrak{S}$ of closed sets consist of the principal ideals generated by

$$\begin{aligned}
&X(X_i\{A_i\}, X_j\{A_j\}) \text{ for } 1 \le i < j \le 4,\\
&X(X_i\{A_i\}, X_j\{\lambda\}, X_k\{\lambda\}),\\
&X(X_i\{\lambda\}, X_j\{A_j\}, X_k\{\lambda\}),\\
&X(X_i\{\lambda\}, X_j\{\lambda\}, X_k\{A_k\}) \text{ for } 1 \le i < j < k \le 4 \text{ and}\\
&X(X_1\{\lambda\}, X_2\{\lambda\}, X_3\{\lambda\}, X_4\{\lambda\}).
\end{aligned}$$

The system of maximal antikeys is the set of coincidence-ideals

$$\begin{aligned}
&X(X_i\{A_i\}, X_j\{\lambda\})\downarrow\cup\{X_I \mid |I| > 1\} \text{ for } i \ne j,\ i, j \in \{1, 2, 3, 4\} \text{ and}\\
&X(X_i\{\lambda\}, X_j\{\lambda\}, X_k\{\lambda\})\downarrow\cup\{X_I \mid |I| > 1\} \text{ for } 1 \le i < j < k \le 4.
\end{aligned}$$

The patterns belonging to $X(X_1\{A_1\}, X_2\{\lambda\})\downarrow\cup\{X_I \mid |I| > 1\}$ are $(+, +, +, -)$ and $(+, +, -, +)$, the edge constraints are $\varphi_1\colon =_1$, $\varphi_2\colon \neq_2$. The same pair of patterns belong to $X(X_2\{A_2\}, X_1\{\lambda\})\downarrow\cup\{X_I \mid |I| > 1\}$, however the edge constraints are $\varphi_1\colon \neq_1$, $\varphi_2\colon =_2$. These two sets of constraints are clearly contradictory, hence by Theorem 33 there exists no Armstrong-instance for $\mathfrak{S}$.

The next example shows that there is a significant difference between the counter-free case and the general case.

Example 36. Consider again $X = X\{X_1(A_1) \oplus X_2(A_2) \oplus X_3(A_3) \oplus X_4(A_4)\}$ and $\mathcal{S}(X)$. Let $\mathfrak{K}$ be the Sperner system of the following closed sets of subattributes. $\mathfrak{K} = \{X(X_1\{\lambda\}, X_2\{\lambda\})\downarrow, X(X_1\{\lambda\}, X_3\{\lambda\})\downarrow, X(X_2\{\lambda\}, X_4\{\lambda\})\downarrow, X(X_3\{\lambda\}, X_4\{\lambda\})\downarrow\}$.

$\mathfrak{K}^{-1}$ consists of $(X(X_2\{A_2\}, X_3\{A_3\})\downarrow$ and $X(X_1\{A_1\}, X_4\{A_4\})\downarrow$ in the counter-free case. It is easy to see that the following three tuples $(X_1 : a, X_2 : b, X_3 : c, x_4 : d), (X_2 : b, X_3 : c), (X_1 : a, x_4 : d)$ form an Armstrong-instance. However, in the general case the maximal antikeys are

$$\mathcal{A}_1 = X(X_2\{A_2\}, X_3\{A_3\})\downarrow \cup \{X_I \mid |I| > 1\}$$

and

$$\mathcal{A}_2 = X(X_1\{A_1\}, X_4\{A_4\})\downarrow \cup \{X_I \mid |I| > 1\}.$$

The pair of patterns determined by $\mathcal{A}_2$ is $(+, -, +, +)$ and $(+, +, -, +)$, while $\mathcal{A}_1$ gives $(-, +, +, +)$ and $(+, +, +, -)$. However, tuples of patterns $(+, +, -, +)$ and $(+, +, +, -)$, respectively, agree on the key $X(X_1\{\lambda\}, X_2\{\lambda\})\downarrow$. Thus, $\mathfrak{K}$ does not have an Armstrong-instance in the case of counter attributes being considered.

4.2 Structural induction?

Most of the proofs about higher-order datamodels exploit structural induction. Some of the constructors allow lifting an Armstrong-instance. Consider the list constructor, for example. Let $X[X']$ be a nested attribute, and let $\mathfrak{K} = \{\mathcal{K}_1, \ldots, \mathcal{K}_m\}$ be a candidate key system in $\mathcal{S}(X')$ that has an Armstrong-instance $r = \{t_1, \ldots, t_s\}$. Then it is easy to see that $\overline{r} = \{[t_1], \ldots, [t_s]\}$ is an Armstrong instance for the candidate key system $\overline{\mathfrak{K}} = \{[\mathcal{K}_1], \ldots, [\mathcal{K}_m]\}$ of $\mathcal{S}(X)$. We use the notation $[\mathcal{K}_i] = \{[K] \mid K \in \mathcal{K}_i\}$.

However, the reverse is obviously not true. Consider $X = X[X'\{X_1(A_1) \oplus X_2(A_2) \oplus X_3(A_3) \oplus X_4(A_4)\}]$ and the candidate key system $\mathfrak{S}$ consisting of $X[X'(X_i\{A_i\}, X_j\{A_j\})]\downarrow$, $1 \le i < j \le 4$, $X[X'(X_i\{A_i\}, X_j\{\lambda\}, X_k\{\lambda\})]\downarrow$, $X[X'(X_i\{\lambda\}, X_j\{A_j\}, X_k\{\lambda\})]\downarrow$, $X[X'(X_i\{\lambda\}, X_j\{\lambda\}, X_k\{A_k\})]\downarrow$ $1 \le i < j < k \le 4$, and $X[X'(X_1\{\lambda\}, X_2\{\lambda\}, X_3\{\lambda\}, X_4\{\lambda\})]\downarrow$. This system consists of non-degenerate keys, thus by Theorem 30 it has an Armstrong-instance. Indeed, $\mathcal{A}$ is a maximal antikey for the candidate key system in Example 35 iff $[\mathcal{A}]$ is a maximal antikey for $\mathfrak{S}$. Since $\mathcal{A}$ is a coincidence ideal, according to property 5(b) of Definition 16 $[\mathcal{A}]$ is a coincidence ideal as well, thus both conditions of Theorem 30 are satisfied. If $\mathfrak{K}$ denotes the candidate key system in Example 35, then $\mathfrak{S} = [\mathfrak{K}]$. $\mathfrak{S}$ has Armstrong-instance, but $\mathfrak{K}$ does not.

This example shows that there is no hope for deciding about Armstrong-instance using structural induction. Another example of the same flavor can be given using the record constructor. Consider $X = X(X'\{X_1(A_1) \oplus X_2(A_2) \oplus X_3(A_3) \oplus X_4(A_4)\}, Y[B])$ and the subattribute lattice $\mathcal{S}(X)$. As before, let $\mathfrak{K}$ denote the candidate key system in Example 35, and let $\mathcal{G}_{\mathcal{K}} = \{X(K, Y[B]) \mid K \in \mathcal{K}\}$. The

Sperner system $\mathfrak{Q} = \{\mathcal{G}_{\mathcal{K}} \mid \mathcal{K} \in \mathfrak{K}\}$ consists of non-degenerate candidate keys. $\mathfrak{Q}^{-1}$ consists of $X(X'\{X_1(A_1) \oplus X_2(A_2) \oplus X_3(A_3) \oplus X_4(A_4)\}, Y[\lambda])\downarrow$ and $X(\mathcal{A}, Y[B])\downarrow$, where $\mathcal{A} \in \mathfrak{K}^{-1}$. These are coincidence ideals, thus by Theorem 30 $\mathfrak{Q}$ has an Armstrong-instance. However, the projection of $\mathfrak{Q}$ to the first component is exactly the system in Example 35.

5 Strong Keys

Keys correspond to ideals of the subattribute lattice with some additional properties. Principal ideals form an important subclass of ideals. Another reason for considering principal ideals is that in the relational datamodel each candidate key that is a closed set is a principal ideal.

Proposition 37. *Let $\mathcal{Y} = Y\downarrow$ be a principal ideal of the subattribute lattice $\mathcal{S}(X)$ of a nested attribute X. Then $\mathcal{Y}$ is closed.*

Proof. One has to show that

$$\mathcal{Y} = \bigcap_{\substack{\mathcal{F} \text{is a coincidence-ideal} \\ \mathcal{Y} \subseteq \mathcal{F}}} \mathcal{F}, \tag{3}$$

or in other words, if $\mathcal{Y} \subset \mathcal{F}$ for a coincidence ideal $\mathcal{F}$ and $Z \in \mathcal{F} \setminus \mathcal{Y}$, then there exists a coincidence ideal $\mathcal{G}$ with $\mathcal{Y} \subset \mathcal{G}$ and $Z \notin \mathcal{G}$. If $\mathcal{Y}$ is not a coincidence ideal itself, then it violates some of the properties of Definition 16. However, a principal ideal can only violate 2(a)-(e). These always give choice that either one or another subattribute must be in a coincidence ideal. Thus to construct $\mathcal{G}$ one only has to avoid adding Z, when there is a choice. Since $\mathcal{S}(X)$ is finite, after finitely many extensions the coincidence ideal $\mathcal{G}$ is obtained.

□

Proposition 37 states that principal ideals are candidate keys. Thus, the next definition is meaningful.

Definition 38. A Sperner system of closed sets of $\mathcal{S}(X)$ is called a *strong* candidate key system if it consists of principal ideals.

Note that in case of record constructor only, that is in the relational datamodel, all keys are strong.

5.1 Record attributes with only one set component

Consider the following restricted record constructor: attribute $Y(Y_1, \ldots, Y_n)$, where $Y_1 = Y_1\{Y_1'\}$, while Y_i is not a set attribute for $i > 1$. Let X be obtained by repeated applications of this constructor. If $\mathcal{K}$ is a degenerate strong candidate key, then it is $X(Y^k\{\lambda\}, \lambda, \ldots, \lambda)\downarrow$ for some k, where $Y^k\{\lambda\}$ stands for $Y\{Y'\{Y''\{\cdots\{\lambda\}\cdots\}$ with k being the nesting depth of set constructors. Let $\mathfrak{S} = \{\mathcal{K}_1, \ldots, \mathcal{K}_m\}$ be a strong candidate key system that contains a degenerate candidate key $\mathcal{K}_1 = X(Y^k\{\lambda\}, \lambda, \dots, \lambda)\downarrow$. By the Sperner property, $\mathcal{K}_i = X(Y^{j_i}\{\lambda\}, Y_2^i, \dots, Y_n^i)\downarrow$ with $j_i < k$ for $i > 1$. Let $k = i_0 > i_1 > \dots > i_p$ be the set of distinct values of j_i's $i = 1, 2, \dots, n$. Furthermore, let X' be the nested attribute $X'(Y_2, Y_3, \dots, Y_n)$, that is the "set-free" component of X. Let $\mathfrak{K}_{i_m}$ be the set of principal ideals in $\mathcal{S}(X')$ defined by $\mathfrak{K}_{i_m} = \{X'(Y_2^i, \dots, Y_n^i)\downarrow | \ j_i = i_m\}$. Since $\mathfrak{S}$ is a Sperner system, $\mathfrak{K}_{i_m}$ is a Sperner system, as well. Also, if $i_f < i_g$, then for all $\mathcal{K} \in \mathfrak{K}_{i_f}$ and $\mathcal{K}' \in \mathfrak{K}_{i_g}$ we have that $\mathcal{K} \not\supseteq \mathcal{K}'$ holds.

Let $\mathcal{A} \in \mathfrak{S}^{-1}$ be a maximal candidate antikey. Since $X(Y^k\{\lambda\}, \lambda, \dots, \lambda)\downarrow$ is a candidate key, every subattribute in $\mathcal{A}$ must have first component of form $Y^h\{\lambda\}$ for some $h < k$. Suppose that $X(Y^h\{\lambda\}, Y_2, \dots, Y_n)$ and $X(Y^{h'}\{\lambda\}, Y_2', \dots, Y_n')$ are two elements of $\mathcal{A}$. Using that $\mathcal{A}$ is an ideal we obtain that

$$X(Y^h\{\lambda\}, \lambda, \dots, \lambda), X(Y^{h'}\{\lambda\}, \lambda, \dots, \lambda) \in \mathcal{A}$$

holds. Clearly $X(Y^h\{\lambda\}, Y_2, \dots, Y_n)$ and $X(Y^{h'}\{\lambda\}, \lambda, \dots, \lambda)$ are reconcilable, and so are $X(Y^{h'}\{\lambda\}, Y_2', \dots, Y_n')$ and $X(Y^h\{\lambda\}, \lambda, \dots, \lambda)$. Thus by property 1 of Definition 16

$$X(Y^h\{\lambda\}, Y_2, \dots, Y_n) \sqcup X(Y^{h'}\{\lambda\}, \lambda, \dots, \lambda) = X(Y^{\max(h,h')}\{\lambda\}, Y_2, \dots, Y_n) \in \mathcal{A}$$

holds. $X(Y^{\max(h,h')}\{\lambda\}, Y_2', \dots, Y_n') \in \mathcal{A}$ is obtained by the same argument. Thus, the first components of the maximal elements of $\mathcal{A}$ are uniquely determined.

Proposition 39. *Let $Y^h\{\lambda\}$ be this unique first component of maximal elements of $\mathcal{A}$. Then $h = i_j - 1$ for some $0 \le j \le p$.*

Proof. Let us assume in contrary, that $i_{j+1} \le h < i_j - 1$ for some j, and let $X(Y^h\{\lambda\}, Y_2, \dots, Y_n)$ be a maximal element of $\mathcal{A}$. Since $Y^h\{\lambda\} > Y^{i_m}\{\lambda\}$ for all $m \ge j+1$, $X'(Y_2, \dots, Y_n)$ cannot be larger than any element of $\mathfrak{K}_{i_m}$. Thus, denoting the projection of $\mathcal{A}$ onto the last $n-1$ components by $\mathcal{A}'$, then it is a candidate antikey (not necessarily maximal) for the (not necessarily Sperner) candidate key system $\mathfrak{K}_{i_{j+1}} \cup \dots \cup \mathfrak{K}_{i_p}$. However, if $\mathcal{A}$ is enlarged by adding $X(Y^{i_j-1}\{\lambda\}, Y_2, \dots, Y_n)$ and elements that must also be added by the ideal property for all maximal element $X(Y^h\{\lambda\}, Y_2, \dots, Y_n)$ of $\mathcal{A}$, then by the observation above, the coincidence ideal obtained remains a candidate antikey, in contradiction with the maximality of $\mathcal{A}$. □

Proposition 39 gives a list of candidate antikeys of $\mathfrak{S}$ that contains all maximal candidate antikeys. For $0 \le j \le p$ take a system of maximal candidate antikeys of $\mathfrak{K}_{i_{j+1}} \cup \dots \cup \mathfrak{K}_{i_p}$ $\mathcal{A}^j_{m_1}, \dots, \mathcal{A}^j_{m_u}$, then extend each with first coordinate $Y^{i_j-1}\{\lambda\}$, finally add all elements of $\mathcal{S}(X)$ that are under some of the obtained maximal elements.

Since $X(Y^k\{\lambda\}, \lambda, \dots, \lambda)\downarrow$ is a candidate key, in an Armstrong-instance any two complex values must have distinct projections onto $X(Y^k\{\lambda\}, \lambda, \dots, \lambda)$ that allows 2^k tuples at most.

6 Numerical Conditions

In the combinatorial investigation of Armstrong-instances of the relational model the following fundamental inequality of comparing the minimal size of Armstrong-instance for a minimal key system $\mathcal{K}$ and the size of the set of maximal antikeys $\mathcal{A} = \mathcal{K}^{-1}$ was proven by Demetrovics and Katona [7].

Lemma 40. *Let $R = (R_1, \ldots, R_n)$ be a relational schema, $\mathcal{K}$ be Sperner system of subsets of R. The minimum number of tuples $s(\mathcal{K})$ in an Armstrong-instance of minimal key system $\mathcal{K}$ satisfies*

$$|\mathcal{K}^{-1}| \leq \binom{s(\mathcal{K})}{2} \quad \textit{and} \quad s(\mathcal{K}) \leq |\mathcal{K}^{-1}| + 1. \tag{4}$$

The analog for the higher-order datamodel was given in [21] for the counter-free case. The same can be stated in the present general case, the similar proof is omitted. Let $\mathcal{S}(X)$ be the subattribute lattice of a nested attribute X. Furthermore let $\mathfrak{K}$ be a Sperner system of closed subsets of $\mathcal{S}(X)$. If $\mathfrak{K}$ has an Armstrong-instance as minimal key system, then $s(\mathfrak{K})$ denotes the minimum number of complex values in an Armstrong-instance of $\mathfrak{K}$. Otherwise, set $s(\mathfrak{K}) = \infty$.

Lemma 41.

$$|\mathfrak{K}^{-1}| \leq \binom{s(\mathfrak{K})}{2} \tag{5}$$

Using Theorem 30 an upper bound can be given in the case when $\mathfrak{K}$ does not contain degenerate keys.

Proposition 42. *Let $\mathcal{S}(X)$ be the subattribute lattice of a nested attribute X, and let $\mathfrak{K}$ be a Sperner system of closed subsets of $\mathcal{S}(X)$. Furthermore, assume that the conditions of Theorem 30 are satisfied. Then*

$$s(\mathfrak{K}) \leq 2|\mathfrak{K}^{-1}|. \tag{6}$$

Proof. The proof of Theorem 30 constructs two complex-valued tuples for each maximal antikey in $\mathfrak{K}^{-1}$.

□

6.1 Only degenerate keys

Having a degenerate key in the candidate key system gives a finite upper bound on the possible number of complex-valued tuples. If the candidate key system consists of only degenerate keys, then a lower bound for the number of maximal antikeys can be established. These two give necessary conditions for the existence of an Armstrong instance via Lemma 41.

Let us consider $\mathcal{S}\left(X\{X_1(A_1) \oplus X_2(A_2) \oplus \cdots \oplus X_n(A_n)\}\right)$ and let $\mathfrak{K}$ be a Sperner system of closed sets, with $\mathfrak{K} = \{X(X_v\{\lambda\} \mid v \in E)\downarrow \mid E \in \mathfrak{E}\}$, where $\mathfrak{E}$ is a Sperner system of subsets of $\{1, 2, \ldots, \}$.

Theorem 43. *Let $X = X\{X_1(A_1) \oplus X_2(A_2) \oplus \cdots \oplus X_n(A_n)$ and let $\mathfrak{K}$ be defined as above. If $\mathfrak{K}$ has an Armstrong-instance, then*

$$\sum_{\substack{V \text{ maximal independent set of} \\ \text{hypergraph } (\{1,2,\ldots,n\},\mathfrak{E})}} \max\left(2^{n-|V|-1} - 1, 1\right) \leq \binom{2^{\min(|E|\colon E\in\mathfrak{E})}}{2}. \tag{7}$$

Proof. In the proof of Proposition 32 it was shown that maximal candidate antikeys in $\mathcal{S}(X\{X_1(A_1) \oplus X_2(A_2) \oplus \cdots \oplus X_n(A_n))$ consist of principal ideals extended with some counter attributes. Thus, the system of candidate maximal antikeys $\mathfrak{K}^{-1}$ consists of coincidence ideals of type $\mathcal{A}_V^{\mathfrak{I}} = X(X_v\{A_v\} \mid v \in V){\downarrow} \cup \{X_I \mid I \in \mathfrak{I}\}$. Here V is a *maximal independent vertex set* of the *hypergraph* (set system) $(\{1,2,\ldots,n\},\mathfrak{E})$ and $\mathfrak{I}$ is a set of at least two-element subsets of $\{1,2,\ldots,n\}$. Indeed, $\mathcal{A}_V^{\mathfrak{I}}$ contains $X(X_v\{\lambda\} \mid v \in E){\downarrow}$ iff $E \subseteq V$. According to property 4(b) of Definition 16 $X(X_v\{A_v\} \mid v \in V){\downarrow}$ has a coincidence ideal extension $\mathcal{A}_V^{\mathfrak{I}}$ for all *non-trivial* partition of I^- into two parts, provided $|I^-| > 1$. Since $|I^-| = n - |V|$, the number of such partitions is $2^{n-|V|-1} - 1$. Thus, the left hand side of (7) is a lower bound of the number of candidate maximal antikeys. A key $X(X_v\{\lambda\} \mid v \in E){\downarrow}$ allows at most $2^{|E|}$ distinct tuples. Applying Lemma 41, (7) follows.

□

7 Conclusions

In the present paper we investigated keys and antikeys in the presence of various constructors in the higher order datamodel. We proved that keys, as well as antikeys, correspond to certain ideals with additional closure properties. These are closed sets, that is intersections of coincidence ideals defined in [21], subsets of the subattribute lattice. We showed that the system of minimal keys correspond to Sperner system of closed sets and exhibited a sufficient condition when such a Sperner system occurs as a system of minimal keys. The candidate key systems not covered by the sufficient condition of Theorem 30 are the ones containing degenerate keys. A characterization when Armstrong-instance exists for such key systems is given in the (possibly) most important special case. Strong keys are also introduced. Some interesting combinatorial problems arose and we are intended to continue our investigations in that direction, as well. Another future direction of research is to refine the existing necessary, or sufficient conditions for Armstrong-instances, preferably to find characterizations in important special cases.

References

[1] Abiteboul, Serge, Buneman, Peter, and Suciu, Dan. *Data on the Web: From Relations to Semistructured Data and XML.* Morgan Kaufmann Publishers, 2000.

[2] Abiteboul, Serge and Hull, Rick. Restructuring hierarchical database objects. *Theoretical Computer Science*, 62(1-2):3–38, 1988.

[3] Abiteboul, Serge, Hull, Rick, and Vianu, Victor. *Foundations of Databases*. Addison-Wesley, 1995.

[4] Armstrong, W. W. Dependency structures of database relationships. *Information Processing*, pages 580–583, 1974.

[5] Brightwell, G. and Katona, G.O.H. A new type of coding theorem. *Studia Scientiarum Mathematicarum Hungarica*, 38:139–147, 2001.

[6] Demetrovics, J. On the equivalence of candidate keys with Sperner systems. *Acta Cybernetica*, 4:247–252, 1979.

[7] Demetrovics, J. and Katona, G.O.H. Extremal combinatorial problems in relational data base. In *Fundamentals of Computing Theory (FCT 1981)*, number 117 in LNCS, pages 110–119. Springer-Verlag, Berlin, 1981.

[8] Demetrovics, J., Katona, G.O.H., and Sali, A. The characterization of branching dependencies. *Discrete Applied Mathematics*, 40:139–153, 1992.

[9] Demetrovics, J., Katona, G.O.H., and Sali, A. Design type problems motivated by database theory. *Journal of Statistical Planning and Inference*, 72:149–164, 1998.

[10] Fagin, Ronald. Horn clauses and database dependencies. *Journal of the Association for Computing Machinery*, 29(4):952–985, 1982.

[11] Fagin, Ronald and Vardi, M. Y. Armstrong databases for functional and inclusion dependencies. *Information Processing Letters*, 16:13–19, 1983.

[12] Ganter, B., Gronau, H.-D. O. F., and Mullin, R. C. On orthogonal double covers of k_n. *Ars Combinatoria*, 37:209–221, 1994.

[13] Hartmann, S. and Link, S. Reasoning about functional dependencies in an abstract data model. *Electronic Notes in Theoretical Computer Science*, 84, 2003.

[14] Hartmann, Sven, Hoffmann, Anne, Link, Sebastian, and Schewe, Klaus-Dieter. Axiomatizing functional dependencies in the higher-order entity relationship model. *Information Processing Letters*, 87(3):133–137, 2003.

[15] Hartmann, Sven, Link, Sebastian, and Schewe, Klaus-Dieter. Reasoning about functional and multi-valued dependencies in the presence of lists. In Seipel, Dietmar and Turull Torres, José María, editors, *Foundations of Information and Knowledge Systems*, volume 2942 of *Springer LNCS*. Springer Verlag, 2004.

[16] Hartmann, Sven, Link, Sebastian, and Schewe, Klaus-Dieter. Weak functional dependencies in higher-order datamodels. In Seipel, Dietmar and Turull Torres, José María, editors, *Foundations of Information and Knowledge Systems*, volume 2942 of *Springer LNCS*. Springer Verlag, 2004.

[17] Hartmann, Sven, Link, Sebastian, and Schewe, Klaus-Dieter. Functional dependencies over XML documents with DTDs. *Acta Cybernetica*, 17(1):153–171, 2005.

[18] Hartmann, Sven, Link, Sebastian, and Schewe, Klaus-Dieter. Axiomatisation of functional dependencies in the presence of records, lists, sets and multisets. *Theoretical Computer Science*, 355:167–196, 2006.

[19] Paredaens, J., De Bra, P., Gyssens, M., and Van Gucht, D. *The Structure of the Relational Database Model*. Springer-Verlag, 1989.

[20] Sali, Attila. Minimal keys in higher-order datamodels. In Seipel, Dietmar and Turull Torres, José María, editors, *Foundations of Information and Knowledge Systems*, volume 2942 of *Springer LNCS*. Springer Verlag, 2004.

[21] Sali, Attila and Schewe, Klaus-Dieter. Counter-free keys and functional dependencies in higher-order datamodels. *Fundamenta Informaticae*, 70:277–301, 2006.

[22] Sali, Attila and Schewe, Klaus-Dieter. Weak functional dependencies on trees with restructuring. Technical Report 4/2006, Massey University, Department of Information Systems, 2006. available from http://infosys.massey.ac.nz/research/rs_techreports.html.

[23] Schewe, Klaus-Dieter and Thalheim, Bernhard. Fundamental concepts of object oriented databases. *Acta Cybernetica*, 11(4):49–85, 1993.

[24] Thalheim, Bernhard. Foundations of entity-relationship modeling. *Annals of Mathematics and Artificial Intelligence*, 6:197–256, 1992.

[25] Thalheim, Bernhard. *Entity-Relationship Modeling: Foundations of Database Technology*. Springer-Verlag, 2000.

Received 6th October 2006