DEPENDENCY CHARACTERIZATIONS FOR ACYCLIC DATABASE SCHEMES

Gösta Grahne and Kari-Jouko Räihä

Department of Computer Science, University of Helsinki
Tukholmankatu 2, SF-00250 Helsinki 25, Finland

Abstract

Acyclic database schemes have attracted a lot of interest because of the nice properties enjoyed by such schemes. Recently some new acyclicity conditions that are strictly stronger than the normal α-acyclicity have been introduced by Fagin. Because of increased requirements, the database schemes in the new classes have some further useful properties that are not shared by α-acyclic schemes. Therefore the new classes have practical relevance.

A database designer may work in terms of attribute sets and data dependencies, and not only in terms of database schemes. Thus it is important to have a characterization for the acyclic schemes of various degree in terms of data dependencies. For α-acyclic schemes such a characterization exists, but for the new classes the question has been open. In this paper we provide characterizations for β-, γ- and Berge-acyclic database schemes. The characterizations can be stated in a simple form: thus they should be useful for the database designer.

1. Introduction

In the relational data model a database scheme for a set of attributes U consists of a set of relation schemes $R=\{R_1,\ldots,R_k\}$, where each R_i, $1 \leq i \leq k$, is a subset of U. Several desirable properties, such as dependency preservation, normal forms and lossless joins have been defined for database schemes [BBG78]. Recently a new important property called acyclicity has gained much attention [BFMY83]. It has been shown that databases that conform to acyclic schemes enjoy some very desirable properties. Thus it is for instance possible in this class of databases to perform joins over several relations in such a way that the number of tuples in the intermediate results is never decreasing. That is, we will never have the inefficient situation where an intermediate result contains a large number of tuples whereas the final result only contains a few tuples. It has even been conjectured that acyclic schemes are the only natural ones, and that a cyclic database scheme is a sign of a design error [FMU82, Sci81].

Very recently it has also been shown that there are several degrees of acyclicity for database schemes [Fag83a, Fag83b], where the previously studied concept of acyclicity is the most general one. The strengthening of the degree of acyclicity also strengthens the related desirable properties.

Although the different degrees of acyclicity and their related properties are quite well known, the same cannot be said about the relationship between acyclicity and data dependencies. If an acyclic (of the most general type) database scheme has the lossless join property, then it is known that the corresponding join dependency is logically equivalent to a set of multivalued dependencies belonging to a class called conflict-free [BFMY83]. Furthermore, the use of a conflict-free set of multivalued dependencies in the standard decomposition process [Lie82] results in a unique database scheme that is acyclic, has the lossless join property, preserves the dependencies and in which each individual scheme is in fourth normal form. This is a very important result since it gives a method for designing acyclic database

This work was supported by the Academy of Finland.

schemes. But for stronger forms of acyclicity nothing is known about the corresponding classes of multivalued dependencies, except that they of course have to be subsets of the conflict-free class.

The problem is solved in this paper for each degree of acyclicity. We will characterize the classes of multivalued dependencies and show the equivalence between each type of acyclic join dependencies and sets of multivalued dependencies. It also follows that the decomposition process results in a stronger form of acyclic database scheme if a more restricted set of multivalued dependencies is used.

Basic notations, definitions and previous results are presented in Section 2 and the new results are found in Section 3. The last section contains the conclusions.

2. Preliminaries

For the definitions of the relational model, multivalued dependencies (MVDs), join dependencies (JDs), hypergraphs etc we refer to [Ull82a, BFMY83]. The following notations will be used. A database scheme is a set $\mathcal{R}=\{R_1,\ldots,R_k\}$ such that $\bigcup_{i=1}^{k} R_i = U$. The join dependency of the scheme is denoted by $*\mathcal{R}$ and the set of multivalued dependencies for U by M. The keys in M are denoted by LHS(M) and they are defined as the set $\{X \mid X \twoheadrightarrow Y \in M\}$. A hypergraph $\mathcal{H}$ is a pair $(\mathcal{N},\mathcal{E})$, where $\mathcal{N}$ is the set of nodes and $\mathcal{E}$ is the set of (hyper)edges. The hypergraph of $\mathcal{R}$ is the pair $(U,\mathcal{R})$, which will be denoted by $\mathcal{R}$ if no confusion arises. Similarly, the hypergraph of LHS(M) is the pair (X,LHS(M)), where X is the set of attributes appearing in LHS(M). For other notations and definitions see the references mentioned above.

A hypergraph $\mathcal{H}$ generates a set M of MVDs in the following way. The dependency $X \twoheadrightarrow Y$ is in M if and only if X and Y are disjoint sets of nodes and Y is the union of some connected components of the hypergraph $\mathcal{H}$-X. We have

Proposition 1 [FMU82]. The set of MVDs implied by a join dependency $*\mathcal{R}$ is exactly the set of MVDs generated by the hypergraph $\mathcal{R}$. □

We will now look into the acyclicity of database schemes, which is defined as the acyclicity of the hypergraphs of the schemes. There are four degrees of acyclicity. They are, in increasing strength, α-acyclicity, β-acyclicity, γ-acyclicity and Berge-acyclicity [Fag83a]. We will also consider cyclic hypergraphs. A hypergraph is θ-cyclic if and only if it is not θ-acyclic, for θ = α, β, γ, Berge. Finally, we say that a JD $*\mathcal{R}$ is θ-acyclic precisely when the hypergraph $\mathcal{R}$ is θ-acyclic.

Definition 1 [BFMY83]. A reduced hypergraph is α-acyclic if and only if all its blocks are trivial. A hypergraph is α-acyclic precisely when its reduction is α-acyclic.

As an example of an α-acyclic hypergraph, Figure 1 shows the hypergraph of the database scheme $\mathcal{R}$ = {ABC, AFE, CDE, ACE}.

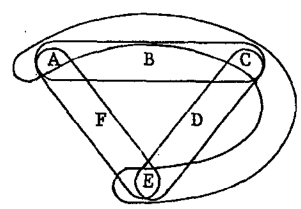

Figure 1. An α-acyclic hypergraph.

An α-acyclic hypergraph can have an α-cyclic subgraph. In Figure 1 the subgraph with nodes {A,B,C,D,E,F} and edges {ABC, CDE, AFE} is α-cyclic.

Given a database scheme $\mathcal{R}$, the next definition gives its join tree.

Definition 2 [BFMY83]. A join tree for a database scheme $\mathcal{R}$ is a tree with set $\mathcal{R}$ of nodes, such that

(i) each edge (R_i,R_j) is labelled by the set of attributes $R_i \cap R_j$, and

(ii) for every pair R_i, R_j $(R_i \neq R_j)$ and for every A in $R_i \cap R_j$, each label of an edge along the unique path between R_i and R_j contains A. This path is called an A-labelled path.

Note that not every database scheme has a join tree, and that there might be several join trees for one database scheme. Using the database scheme {ABC, AFE, CDE, ACE} we can construct the join tree shown in Figure 2.

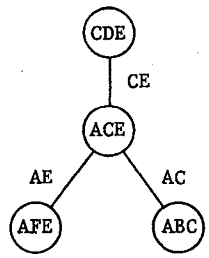

Figure 2. A join tree for {ABC, AFE, CDE, ACE}.

A join tree corresponds to a set of multivalued dependencies in the following way.

Definition 3 [BFMY83]. An MVD corresponding to an edge (R_i,R_j) of a join tree T for a database scheme R is $R_i \cap R_j \to\to N - R_i \cap R_j$, where N is the union of the nodes in one of the two subtrees that are obtained by deleting the edge (R_i,R_j) from T. The set of MVDs corresponding to a join tree is the union of the MVDs corresponding to the edges of the tree.

For instance, a set of MVDs corresponding to the join tree in Figure 2 is {AE→→F, AC→→B, CE→→D}.

We next define some notions describing the effect of a dependency set on attribute sets.

Definition 4. Let M be a set of MVDs.

(a) A set X in LHS(M) separates the sets P and Q whenever there exist sets P' and Q' such that $P' \cap Q' = \emptyset$, $P \subseteq P'$, $Q \subseteq Q'$, $X \to\to P'$, and $X \to\to Q'$. We say that M separates P and Q precisely when some X in LHS(M) separates P and Q.

(b) A dependency X→→Y in M splits a set V if $V \cap Y \neq \emptyset$ and $(U-XY) \cap V \neq \emptyset$. The set M splits V precisely when some X→→Y in M splits V.

We will need the following property of splittings.

Proposition 2 [BFMY83]. A set V is split by a set M of MVDs if and only if V is split by its closure M^+. □

It is easy to see that we have

Corollary. A set M of MVDs separates P and Q if and only if M^+ separates P and Q. □

Definition 5 [BFMY83]. A set M of MVDs is conflict-free if

(i) no key in M is split by M, and

(ii) for all keys X and Y in M, $DEP(X) \cap DEP(Y) \subseteq DEP(X \cap Y)$.

For example, the set {AE→→F, AC→→B, CE→→D} is conflict-free since the keys are not split, and since condition (ii) of the definition is trivially satisfied because the dependency bases for the keys have no common elements.

Our next two results from [BFMY83] relate the concepts defined above.

Proposition 3 [BFMY83]. A hypergraph R has a join tree if and only if R is α-acyclic. □

Proposition 4 [BFMY83]. The set of MVDs corresponding to the join tree of an α-acyclic hypergraph R is conflict-free and equivalent to $*R$. □

In the example that we have elaborated we saw that the database scheme {ABC, AFE, CDE, ACE} has an α-acyclic hypergraph (Figure 1) and a join tree (Figure 2). It is easy to verify that *{ABC,AFE,CDE, ACE} and {AE→→F, AC→→B, CE→→D} are equivalent. The algorithm for inferring MVDs from a JD is essentially the same that was used for generating MVDs from a hypergraph, and an algorithm for the other direction can be found in [Ull82a].

The rest of this section will be used to define stronger degrees of acyclicity. We begin with β-acyclicity.

Definition 6 [Fag83a]. A weak β-cycle in a hypergraph R is a sequence $(S_1,x_1,\ldots,S_m,x_m,S_{m+1})$ such that

(i) $x_1,\ldots,x_m$ are distinct nodes of R;

(ii) $S_1,\ldots,S_m$ are distinct edges of R, and $S_{m+1}=S_1$;

(iii) $m \geq 3$, that is, there are at least three edges involved; and

(iv) x_i is in S_i and S_{i+1} $(1 \leq i \leq m)$, and in no other S_j.

A hypergraph is β-acyclic if and only if it has no weak β-cycle.

The α-acyclic hypergraph of Figure 1 is β-cyclic since the sequence (ABC,A,AFE,E,CDE,C,ABC) is a weak β-cycle (the edges are underlined). We see that A is only in ABC and AFE, E is only in AFE and

CDE, and C is only in CDE and ABC.

For γ-acyclicity we make the following definition.

Definition 7 [Fag83a]. A γ-cycle in a hypergraph R is a sequence $(S_1,x_1,\ldots,S_m,x_m,S_{m+1})$ such that

(i) $x_1,\ldots,x_m$ are distinct nodes of R;
(ii) $S_1,\ldots,S_m$ are distinct edges of R, and $S_{m+1}=S_1$;
(iii) $m \geq 3$;
(iv) x_i is in S_i and S_{i+1} $(1 \leq i \leq m)$; and
(v) if $1 \leq i \leq m$, then x_i is in no S_j except S_i and S_{i+1}.

A hypergraph is γ-acyclic if and only if it has no γ-cycle.

Figure 3 shows a γ-cyclic hypergraph. Note that this hypergraph is β-acyclic.

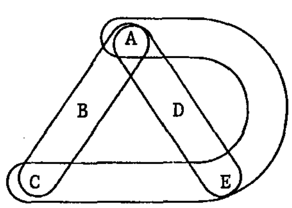

Figure 3. A γ-cyclic hypergraph.

The γ-cycle in the hypergraph is given by the sequence (ABC,C,ACE,E,ADE,A,ABC). Now C is only in ABC and ACE and E is only in ACE and ADE. Since A is in all three edges the γ-cycle is not a weak β-cycle.

Finally we give the definition for Berge-acyclicity.

Definition 8 [Fag83a]. A Berge-cycle in a hypergraph R is a sequence $(S_1,x_1,\ldots,S_m,x_m,S_{m+1})$ such that

(i) $x_1,\ldots,x_m$ are distinct nodes of R;
(ii) $S_1,\ldots,S_m$ are distinct edges of R, and $S_{m+1}=S_1$;
(iii) $m \geq 2$, that is, there are at least 2 edges involved; and
(iv) x_i is in S_i and S_{i+1} $(1 \leq i \leq m)$.

A hypergraph is Berge-acyclic if and only if it has no Berge-cycle.

An example of a Berge-cyclic hypergraph is shown in Figure 4. This hypergraph is γ-acyclic.

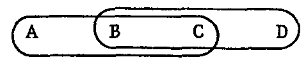

Figure 4. A Berge-cyclic hypergraph.

The Berge-cycle consists of the sequence (ABC,B,BCD,C,ABC). The sequence is not a γ-cycle since only two edges are involved. The following proposition also holds.

Proposition 5 [Fag83a]. If some pair of edges of a hypergraph has two or more nodes in common, the hypergraph is Berge-cyclic. □

We have seen in the examples that there are hypergraphs that are α-acyclic but β-cyclic, β-acyclic but γ-cyclic, and γ-acyclic but Berge-cyclic. The general idea is that going from α-acyclicity to β-, γ- and Berge-acyclicity strengthens the requirements. This can also be seen from the definitions. The following result holds.

Proposition 6 [Fag83a]. Berge-acyclicity ⇒ γ-acyclicity ⇒ β-acyclicity ⇒ α-acyclicity. None of the reverse implications holds. □

3. Characterizations

We start with a technical lemma that is useful for proving the characterization of β-acyclicity.

Lemma. Let R be an α-acyclic hypergraph and M the set of MVDs corresponding to the join tree of R. If the hypergraph LHS(M) has a weak β-cycle $(S_1,x_1,\ldots,S_m,x_m,S_{m+1})$, then none of the paths in the join tree of R can contain three edges whose distinct labels belong to $\{S_1,\ldots,S_m\}$.

Proof. Recall that by the construction of M, each element of LHS(M) (in particular, each S_i) is the label of an edge in the join tree of R. Suppose to the contrary that an edge labelled with S_k belongs to the path that connects the edges labelled with S_i and S_j (see Figure 5; the ancestor relation of the nodes is irrelevant for our purposes).

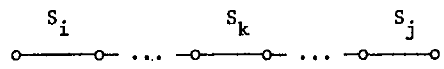

Figure 5. A path in the join tree.

We will use induction on $|j-i|$ to show that the situation is not possible.

Basis. $|j-i| = 1$. Without loss of generality, suppose that $j = i+1$. By the definition of a weak β-cycle, x_i is in S_i and S_{i+1} and in no other S_k. The definition of a join tree implies that every edge on the path from the S_i-edge to the S_j-edge (= S_{i+1}-edge) is labelled with x_i. In particular, $x_i \in S_k$: a contradiction.

Induction step. Suppose that the claim holds for any S_h and $S_{h'}$ with $1 \leq |h'-h| \leq p$; we will show that it holds for S_i and $S_j = S_{i+p+1}$.

By the induction hypothesis, the path from the S_{i+p}-edge to the S_j-edge cannot contain the S_k-edge (since $j-(i+p) = 1 \leq p$). This leaves the possibilities shown in Figure 6.

In either case, the S_k-edge belongs to the path from the S_i-edge to the S_{i+p}-edge. This contradicts the induction hypothesis, since $(i+p)-i = p$. □

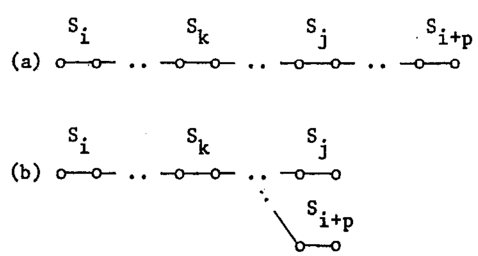

Figure 6. Refinements of the path in Figure 5.

A set of MVDs equivalent to the β-cyclic join dependency *{ABC, AFE, CDE, ACE} (see Figure 1) is {AE↠F, AC↠B, CE↠D}. A closer look at the keys of this set reveals a weak β-cycle given by the sequence (AE,A,AC,C,CE,E,AE). It turns out that this is the source of the β-cyclicity of the JD. We have the following general result.

Theorem 1. A set of multivalued dependencies M' is equivalent to a β-acyclic join dependency *$\mathcal{R}$ if and only if M' has a cover M with the following properties:

(a) M is conflict-free, and

(b) LHS(M) is β-acyclic.

Proof. *Only if*. Suppose that M' ≡ *$\mathcal{R}$ where $\mathcal{R}$ is β-acyclic; we must show that M' has a cover M satisfying (a) and (b). By Proposition 6, $\mathcal{R}$ is also α-acyclic, and thus $\mathcal{R}$ has a join tree (by Proposition 3). Let M be the set of MVDs that corresponds to the join tree of $\mathcal{R}$. By Proposition 4, M is conflict-free and M ≡ *$\mathcal{R}$. Therefore M is a cover of M', and it remains to show that (b) holds for M.

Suppose not. Then there exists a weak β-cycle $(S_1,x_1,\ldots,S_m,x_m,S_{m+1})$ of LHS(M) such that

(i) $x_1,\ldots,x_m$ are distinct nodes of the hypergraph LHS(M);

(ii) $S_1,\ldots,S_m$ are distinct edges of LHS(M), and $S_{m+1}=S_1$;

(iii) $m \geq 3$; and

(iv) x_i is in S_i and S_{i+1} $(1 \leq i \leq m)$, and in no other S_j.

By the construction of M, each S_i is the label of an edge (R_i,R_i') of the join tree of $\mathcal{R}$ so that $S_i = R_i \cap R_i'$. The join tree must have a unique path that connects any two distinct edges. Let $(R_{i,0},R_{i,1},\ldots,R_{i,n_i},R_{i,n_i+1})$ be the path that connects the edges labelled with S_i and S_{i+1}; here $n_i \geq 1$, $R_{i,0} \cap R_{i,1} = S_i$, $R_{i,n_i} \cap R_{i,n_i+1} = S_{i+1}$, and $(R_{i,j},R_{i,j+1})$ is an edge of the join tree for $0 \leq j \leq n_i$.

We claim that $(R_{1,0},x_1,\ldots,R_{m,0},x_m,R_{m+1,0})$ is a weak β-cycle of $\mathcal{R}$ (the x_i:s are the same as in the weak β-cycle of LHS(M)). Since we assumed that $\mathcal{R}$ is β-acyclic, this yields a contradiction and proves the only if -part of the claim. We will deal with the four conditions one at a time.

(i) We must show that $x_1,\ldots,x_m$ are distinct nodes of $\mathcal{R}$. This follows immediately from the facts that $x_1,\ldots,x_m$ are distinct nodes of LHS(M) and $\cup$LHS(M) $\subseteq \cup\mathcal{R}$.

(ii) Since $R_{1,0},\ldots,R_{m,0}$ are nodes of the join tree, they are edges of $\mathcal{R}$, and $R_{m+1,0}=R_{1,0}$ by the construction (recall that $S_{m+1}=S_1$). The fact that the $R_{i,0}$-sets are distinct follows easily from properties (iii) and (iv) proved below: because x_i is in $R_{i,0}$ and $R_{i+1,0}$ and in no other $R_{j,0}$, $R_{i,0}$ is distinct from all the other $R_{j,0}$-sets, with the possible exception of $R_{i+1,0}$. But x_{i+1} is in $R_{i+1,0}$ and not in $R_{i,0}$ (since $m\geq 3$); therefore $R_{i,0}$ is also distinct from $R_{i+1,0}$. Since this holds for each i ($1\leq i\leq m$), all the $R_{i,0}$-sets are distinct.

(iii) $m\geq 3$. Obvious by the construction.

(iv) We claim that x_i is in $R_{i,0}$ and $R_{i+1,0}$ ($1\leq i\leq m$), and in no other $R_{j,0}$. To see that $x_i \in R_{i,0}\cap R_{i+1,0}$, simply recall that $x_i\in S_i\cap S_{i+1} = (R_{i,0}\cap R_{i,1}) \cap (R_{i+1,0}\cap R_{i+1,1}) \subseteq R_{i,0}\cap R_{i+1,0}$. Suppose then that $x_i\in R_{j,0}$, for some $j\neq i,i+1$. Because $x_i\in R_{i,0}$, property (ii) of Definition 2 implies that x_i belongs to the label of every edge on the path from $R_{i,0}$ to $R_{j,0}$. Since $x_i\notin S_j$, the edge labelled with S_j does not belong to this path. Thus we must have one of the situations shown in Figure 7.

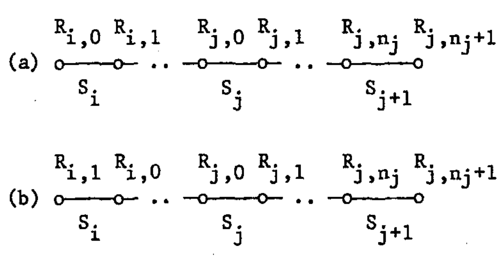

Figure 7. Two possible paths from $R_{i,0}$ to R_{j,n_j+1}.

In either case, we see that the edge labelled with S_j belongs to the path that connects edges S_i and S_{j+1}, contradicting the Lemma.

Therefore x_i cannot be in $R_{j,0}$, concluding the proof of the only if -part.

If. Suppose that $M'\equiv M$, where M satisfies (a) and (b). We must show that there exists a β-acyclic join dependency $*\mathcal{R}$ such that $M'\equiv *\mathcal{R}$.

Since M is conflict-free by property (a), we can apply Lien's decomposition algorithm [Lie82] to M to produce an α-acyclic database scheme $\mathcal{R}$ such that $M\equiv *\mathcal{R}$, and thus also $M'\equiv M\equiv *\mathcal{R}$. If $\mathcal{R}$ is β-acyclic, we are done. Suppose therefore that $\mathcal{R}$ is β-cyclic, i.e. there exists a weak β-cycle $(R_1,x_1,\ldots,R_m,x_m,R_{m+1})$ in $\mathcal{R}$. Because $\mathcal{R}$ is α-acyclic, it has a join tree (by Proposition 3). For each $1\leq i\leq m$, let S_i be the label of the first edge on the unique path from R_i to R_{i+1}, and define $S_{m+1}=S_1$. We will show that $(S_1,x_1,\ldots,S_m,x_m,S_{m+1})$ is a weak β-cycle of LHS(M). This contradicts property (b) of M and proves the claim.

(i) We must show that $x_1,\ldots,x_m$ are distinct nodes of the hypergraph LHS(M). It is clear that they are distinct; we must only show that for $1\leq i\leq m$, $x_i\in X$ for some $X\in$ LHS(M).

Consider R_i and R_{i+1}. In the decomposition tree produced by Lien's algorithm both R_i and R_{i+1} are leaves. Let Z be their lowest common ancestor in the decomposition tree, and let $X\twoheadrightarrow Y_1 \mid Y_2 \mid \ldots \mid Y_k$ be the dependencies used for decomposing Z. Since only dependencies of M are used in the decomposition, $X\in$ LHS(M). During the decomposition step Z is replaced by $X(Y_1\cap Z)$, ..., $X(Y_k\cap Z)$. Suppose that $R_i \subseteq X(Y_p\cap Z)$ and $R_{i+1} \subseteq X(Y_q\cap Z)$. Then $R_i\cap R_{i+1}\subseteq (X(Y_p\cap Z)) \cap (X(Y_q\cap Z)) = X\cup((Y_p\cap Z)) \cap (Y_q\cap Z)) = X$, because Y_p and Y_q are distinct members of DEP(X) and thus their intersection is empty.

Because M is conflict-free, no keys are split in the decomposition process [Lie82]. It follows that since $R_i\cap R_{i+1}\subseteq X$, either $R_i\cap R_{i+1}=\emptyset$ or $R_i\cap R_{i+1}=X$. The intersection cannot be empty, since it contains x_i; thus $x_i\in R_i\cap R_{i+1} = X\in$ LHS(M).

(ii) $S_1,\ldots,S_m$ must be distinct edges of LHS(M). Each S_i is the label of some edge (R_i,R_i') in the join tree. Exactly as in (i) above, we see that $S_i = R_i \cap R_i' \in$ LHS(M). The fact that the S_i-sets are distinct follows immediately from properties (iii) and (iv) below.

(iii) $m \geqslant 3$. Obvious by the construction.

(iv) Finally, we must show that x_i is in S_i and S_{i+1}, and in no other S_j. We know that $x_i \in R_i \cap R_{i+1}$. By the definition of a join tree, x_i belongs to the label of every edge on the path from R_i to R_{i+1} in the join tree. In particular, $x_i \in S_i$ (the label of the first edge on the path).

Like in the proof of condition (iv) for the only if -part, we see that S_{i+1} must in fact be the label of the last edge on the path from R_i to R_{i+1}. Otherwise the path from R_i to R_{i+2} (or R_2 if i=m) would contain the edges labelled with S_i, S_{i+1}, and S_{i+2}, contradicting the Lemma. Therefore we have also $x_i \in S_{i+1}$.

Because $x_i \notin R_j$ when $j \neq i,i+1$ (by the definition of the weak β-cycle in $\mathcal{R}$), we also have $x_i \notin R_j \cap T$ for any $T \in \mathcal{R}$ when $j \neq i,i+1$. In particular, $x_i \notin S_j$ when $j \neq i,i+1$. □

γ-cycles in join dependencies are not directly generated by cycles in the keys of the set of equivalent MVDs. The keys however play an important role in this context also, as can be seen from the following theorem.

Theorem 2. A set of multivalued dependencies M' is equivalent to a γ-acyclic join dependency $*\mathcal{R}$ if and only if M' has a cover M with the following properties:

(a) M is conflict-free, and

(b) for any two distinct keys X and Y in LHS(M) whose intersection is non-empty, $X \cap Y \in$ LHS(M) and $X \cap Y$ separates X-Y and Y-X.

To illustrate the intuitive idea behind the conditions we will give an example before proving the theorem. The JD equivalent to {AB→→C, AB→→D, AE→→F, AE→→G} is shown as a hypergraph in Figure 8(a). The set of MVDs does not satisfy condition (b) of the theorem, since the intersection of AB and AE is non-empty and no key A belongs to the set. The hypergraph of Figure 8(a) is γ-cyclic. By adding the dependency A→→BCD to the set of MVDs we get a set that satisfies the conditions of Theorem 2. Note in particular that A→→BCD separates B and E. The equivalent JD is now *{ABC, ABD, AEF, AEG}, which is γ-acyclic. The hypergraph is shown in Figure 8(b). One can note that adding the dependency A→→BCD to the set of MVDs will make the set AEB unnecessary in the equivalent JD. Thus the γ-cycle will be removed.

(a)

(b)

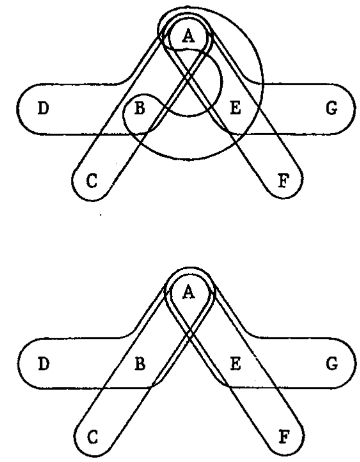

Figure 8. Hypergraphs for {ABC, ABD, AEF, AEG, AEB} and {ABC, ABD, AEF, AEG}.

Proof of Theorem 2. *Only if.* Suppose that $M' \equiv *\mathcal{R}$ where $\mathcal{R}$ is γ-acyclic; we must show that M' has a cover M satisfying (a) and (b). Like in the proof of Theorem 1, let M be the set of MVDs that corresponds to the join tree of $\mathcal{R}$. Then (a) is immediate.

To see that (b) holds, we will first show that $X \cap Y \in$ LHS(M) whenever $X \cap Y \neq \emptyset$ for distinct X and Y in LHS(M). Suppose to the contrary that $\emptyset \neq X \cap Y \notin$ LHS(M) for some X and Y in LHS(M). In particular, $X \cap Y \neq X$; therefore $X \cap Y \neq \emptyset$ and $X - Y \neq \emptyset$.

Consider the join tree of R, and the path $(R_0,R_1,\dots,R_k)$ that connects the edges labelled with X and Y; i.e $R_0 \cap R_1 = X$ and $R_{k-1} \cap R_k = Y$. By Definition 2, the path from R_0 to R_k is an $X \cap Y$-labelled path. Let (R_{i-1},R_i) be the last edge (going from R_0 to R_k) whose label contains an attribute of X-Y. Such an edge (with $i<k$) must belong to the path, since $R_{k-1} \cap R_k = Y$ does not contain attributes of X-Y. Let A be an attribute of $X \cap Y$ and B an attribute of $(X-Y) \cap (R_{i-1} \cap R_i)$.

Consider then the edge (R_i,R_{i+1}). We know that $X \cap Y \subseteq R_i \cap R_{i+1}$. The containment must be strict: otherwise $X \cap Y$ would be the label of an edge in the join tree, contrary to the assumption that $X \cap Y \notin$ LHS(M). Therefore $(R_i \cap R_{i+1}) - (X \cap Y) \neq \emptyset$; let C be an arbitrary attribute of $(R_i \cap R_{i+1}) - (X \cap Y)$.

Combining the properties of the attributes defined above, we see that $A \in R_0$, $A \in R_i$, and $A \in R_{i+1}$; $B \in R_0$, $B \in R_i$, and $B \notin R_{i+1}$; and $C \notin R_0$, $C \in R_i$, and $C \in R_{i+1}$. It is now straightforward to verify that $(R_0,B,R_i,C,R_{i+1},A,R_0)$ satisfies all the conditions of Definition 7. Thus it is a γ-cycle and R is γ-cyclic; a contradiction. Therefore $X \cap Y \in$ LHS(M).

It remains to show that $X \cap Y$ separates X-Y and Y-X. If $X \subseteq Y$ or $Y \subseteq X$, this holds trivially. Suppose then that $X-Y \neq \emptyset$ and $Y-X \neq \emptyset$. The preceding discussion shows that the path that connects the edges labelled with X and Y must contain an edge labelled with $X \cap Y$. Then M contains the dependency $X \cap Y \rightarrow\rightarrow N - X \cap Y$, where N is the set of attributes in one of the components obtained from the join tree by deleting the $X \cap Y$-edge. Since X-Y and Y-X belong to different components, this dependency separates X-Y and Y-X, completing the proof of the only if -part.

If. Suppose that $M' \equiv M$, where M satisfies (a) and (b). We must show that there exists a γ-acyclic join dependency $*R$ such that $M' \equiv *R$.

We begin by showing that (a) and (b) imply that LHS(M) is β-acyclic. Suppose not; then LHS(M) has a weak β-cycle $(S_1,x_1,\dots,S_m,x_m,S_{m+1})$. By the definition, $\{x_i,x_{i+1}\} \subseteq S_{i+1}$ for $1 \leq i \leq m$. Consider an arbitrary S_j with $1 \leq j \leq m$. Because $S_j \cap S_{j+1} \neq \emptyset$, condition (b) implies that $S_j \cap S_{j+1}$ separates $S_j - S_{j+1}$ and $S_{j+1} - S_j$. Because $x_{j-1} \in S_j - S_{j+1}$ and $x_{j+1} \in S_{j+1} - S_j$ (where $x_{m+1} = x_1$), $S_j \cap S_{j+1}$ separates x_{j-1} and x_{j+1}. Thus all the nodes $\{x_1,\dots,x_{j-1}, x_{j+1},\dots,x_m\}$ cannot belong to the same member of DEP$(S_j \cap S_{j+1})$. It follows that there must exist two consecutive nodes x_h and x_{h+1} that are separated by $S_j \cap S_{j+1}$. Because $\{x_h x_{h+1}\} \subseteq S_h$, $S_j \cap S_{j+1}$ splits S_h. This contradicts property (a), and therefore LHS(M) must be β-acyclic.

We can again apply Lien's decomposition algorithm to M to produce a database scheme R. By Theorem 1, R is β-acyclic. If R is γ-acyclic, we are done. Suppose therefore that R is γ-cyclic; we will complete the proof by deriving a contradiction of property (b).

Let $(R_1,x_1,\dots,R_n,x_n,R_{n+1})$ be the shortest γ-cycle in R. It is easily seen that n must equal 3. For suppose that $n \geq 4$. We know that $x_n \in R_i$ for some $1<i<n$, otherwise the cycle would be a weak β-cycle. Since $n \geq 4$, either $n-i+1 \geq 3$ or $i \geq 3$, or else we would have $n < n+1 = (n-i+1)+i \leq 2+2 = 4$. But then either $(R_1,x_1,\dots,R_i,x_n,R_1)$ or $(R_i,x_i,\dots,R_n,x_n,R_i)$ is a γ-cycle, contradicting the minimality of n. Therefore the shortest γ-cycle of R must be of the form $(R_1,x_1,R_2,x_2,R_3,x_3,R_1)$, where the intersections of the sets are shown in Figure 9.

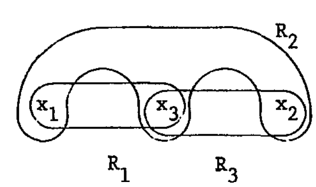

Figure 9. The shortest γ-cycle.

Consider the set of MVDs M'' generated by R. Let $P \rightarrow\rightarrow Q$ be an arbitrary element of M''. If $x_1 \notin P$ and $x_2 \notin P$, then x_1 and x_2 are connected by $R_2 - P$; therefore Q cannot separate x_1 and x_2. If $x_1 \in P$ or $x_2 \in P$, Q cannot separate x_1 and x_2 by the definition of separation. Since $P \rightarrow\rightarrow Q$ was arbitrary, it follows that M'' does not separate x_1 and x_2. By the

Corollary of Proposition 2, M''^+ does not separate x_1 and x_2 either. By Proposition 1, $M''^+ \equiv *R \equiv M$; therefore M does not separate x_1 and x_2.

On the other hand, both $R_1 \cap R_2$ and $R_2 \cap R_3$ belong to LHS(M) (this can be seen as in the proof of Theorem 1). Their intersection is non-empty, since it contains x_3. Condition (b) then implies that $(R_1 \cap R_2) \cap (R_2 \cap R_3)$ separates $R_1 \cap R_2 - R_2 \cap R_3$ and $R_2 \cap R_3 - R_1 \cap R_2$. In particular, $(R_1 \cap R_2) \cap (R_2 \cap R_3)$ separates x_1 and x_2. This contradicts the above observation that M does not separate x_1 and x_2. The counter assumption is therefore false and R is γ-acyclic. □

For Berge-acyclic join dependencies Proposition 5 gives a good hint: no key in the equivalent set of MVDs should be of a size greater than 1. This is stated in condition (a) of Theorem 3. Since key-splitting is prohibited by this condition, the assumption of conflict-freedom can be relaxed into condition (b) of the theorem.

Theorem 3. A set of multivalued dependencies M' is equivalent to a Berge-acyclic join dependency $*R$ if and only if M' has a cover M with the following properties:

(a) $|X| \leq 1$ for any X in LHS(M); and

(b) if $A \twoheadrightarrow Y$ and $B \twoheadrightarrow Y$ for some distinct A and B in LHS(M), then $\emptyset \twoheadrightarrow Y$.

Proof. *Only if*. Suppose that $M' \equiv *R$ where R is Berge-acyclic; we must show that M' has a cover M satisfying (a) and (b). Let again M be the set of MVDs that corresponds to the join tree of R.

Suppose that $|X| > 1$ for some X in LHS(M). Since $X = R_1 \cap R_2$ for some R_1, R_2 in R, we have $|R_1 \cap R_2| > 1$. Proposition 5 then implies that R is Berge-cyclic, contradicting our assumption. Thus (a) holds. Condition (b) follows easily: we know that M is conflict-free (by Proposition 4), and (b) is just a consequence of the condition $DEP(X) \cap DEP(Y) \subseteq DEP(X \cap Y)$ under restriction (a).

If. Suppose that $M' \equiv M$, where M satisfies (a) and (b). We must show that there exists a Berge-acyclic join dependency $*R$ such that $M' \equiv *R$.

It is easily seen that conditions (a) and (b) of the theorem imply conditions (a) and (b) of Theorem 2. Let R be the scheme produced by applying Lien's decomposition algorithm to M. By the proof of Theorem 2, R is γ-acyclic.

Suppose then that R is Berge-cyclic, and let $(S_1, x_1, \ldots, S_m, x_m, S_{m+1})$ be the shortest Berge-cycle in R. If $m \geq 3$, each x_i can only belong to S_i and S_{i+1}, and to no other S_j (otherwise we could delete from the cycle the part between S_j and S_i and obtain a shorter Berge-cycle). But then $(S_1, x_1, \ldots, S_m, x_m, S_{m+1})$ would be a weak β-cycle, contradicting the fact that R must be γ-acyclic (and thus β-acyclic).

Therefore $m = 2$, and the shortest Berge-cycle is of the form $(S_1, x_1, S_2, x_2, S_1)$. But now $\{x_1, x_2\} \subseteq S_1 \cap S_2$. Like in the proof of Theorem 1, we see that $S_1 \cap S_2 \in$ LHS(M). This contradicts condition (a) and proves the claim. □

4. Conclusions

We have characterized acyclic join dependencies of various degree in terms of equivalent sets of multivalued dependencies. The characterizations emphasize the role of the left sides of the dependencies. This is not surprising considering the importance of left sides in the decomposition process ([Lie82], [GrR83]).

The characterizations can be stated in a relatively simple form. Thus they should be helpful to a database designer aiming at an acyclic scheme of a certain degree. For instance, stronger forms of acyclicity can be obtained by splitting some attributes which have been recognized as critical by the characterizations, or by adding some MVDs like in the example in connection with Theorem 2. These questions, however, require further studies. One can note that similar methods have been proposed for obtaining α-acyclicity [Ull82b, BeK83] and independence [Sci83].

References

BBG78 C. Beeri, P.A. Bernstein & N. Goodman, A sophisticate's introduction to database normalisation theory. Proc. of the Fourth International Conference on Very Large Data Bases, September 1978, 113-124.

BFMY83 C. Beeri, R. Fagin, D. Maier & M. Yannakakis, On the desirability of acyclic database schemes. J. ACM 30, 3 (July 1983), 479-513.

BeK83 C. Beeri & M. Kifer, Elimination of intersection anomalies from database schemes. Proc. of the Second ACM SIGACT-SIGMOD Symposium on Principles of Database Systems, March 1983, 340-351.

Fag83a R. Fagin, Degrees of acyclicity for hypergraphs and relational database schemes. J. ACM 30, 3 (July 1983), 514-550.

Fag83b R. Fagin, Acyclic database schemes (of various degrees): a painless introduction. CAAP'83, 8th Colloquium on Trees in Algebra and Programming, G. Ausiello & M. Protasi (Eds.), Springer-Verlag, 1983, 65-89.

FMU82 R. Fagin, A.O. Mendelzon & J.D. Ullman, A simplified universal relation assumption and its properties. ACM Transactions on Database Systems 2, 3 (September 1982), 343-360.

GrR83 G. Grahne & K.-J. Räihä, Database decomposition into fourth normal form. Proc. of the Ninth International Conference on Very Large Data Bases, M. Schkolnick & C. Thanos (Eds.), October-November 1983, 186-196.

Lie82 Y.E. Lien, On the equivalence of database models. J. ACM 29, 2 (April 1982), 333-362.

Sci81 E. Sciore, Real-world MVD's. Proc. of the ACM SIGMOD 1981 Conference on Management of Data, April 1981, 121-132.

Sci83 E. Sciore, Improving database schemes by adding attributes. Proc. of the Second ACM SIGACT-SIGMOD Symposium on Principles of Database Systems, March 1983, 340-351.

Ull82a J.D. Ullman, Principles of Database Systems (Second Edition). Computer Science Press, Potomac, Md., 1982.

Ull82b J.D. Ullman, The U.R. strikes back. Proc. of the ACM Symposium on Principles of Database Systems, March 1982, 10-22.