Theoretical Computer Science 35 (1985) 165-177
North-Holland

ON THE EXISTENCE OF ACYCLIC VIEWS IN A DATABASE SCHEME

Giorgio AUSIELLO and Alessandro d'ATRI
Dipartimento di Informatica e Sistemistica, Universita' "La Sapienza", Roma, Italy

Marina MOSCARINI
Istituto di Analisi dei Sistemi ed Informatica del CNR, Roma, Italy

Communicated by D. Perrin
Received June 1984

Abstract. The importance of acyclic database schemes in relational database theory has been pointed out in various contributions in the literature. Unfortunately, the realm of interest which is captured by the database scheme is often intrinsically cyclic; therefore, we are faced with the problem of finding acyclic views on such a scheme. In this paper we consider three kinds of acyclicity, called α-, γ- and Berge-acyclicity by Fagin (1983), and we approach the problem of the existence of acyclic views in a database scheme. We show that the problem of deciding whether there exists a Berge-, γ-, or α-acyclic view in a general database scheme is NP-complete and that the problem of deciding whether there exists a Berge- or γ-acyclic view on an α-acyclic scheme is also computationally intractable. On the other side, if the given database scheme is γ-acyclic, the problem of deciding the existence of a Berge-acyclic view may be solved by means of efficient algorithms which may also be used to find an acyclic view which involves the minimum number of relations.

Key words. Relational database, universal relation, view, hypergraph, acyclicity, NP-completeness.

C.R. Categories. G.2.2; F.2.2; H.2.1; H.2.4.

1. Introduction

A central problem in relational database research in the last few years has been the problem of investigating properties deriving from the 'structure' of a database scheme. In particular, acyclic database schemes have been defined and it has been proved that such schemes enjoy several desirable properties related to database design and use (see, among others, [2, 8, 9, 10, 12]). A survey on these topics can be found in [7].

Among acyclic database schemes, classes based on different structural properties have been introduced. A particularly relevant role is played by the so-called α-acyclic schemes [2], by γ-acyclic schemes [8] and by Berge-acyclic [4] schemes.

Due to the desirability of their properties, several authors have investigated the problem of designing acyclic database schemes.

In [5], a step-by-step methodology is provided that allows to generate acyclic schemes by adding a relation scheme at a time and by constraining the way in which the new relation scheme is connected to the pre-existing database scheme.

A different approach leads the designer to produce the acyclic scheme by modifying a pre-existing cyclic scheme. Goodman and Shmueli [13, 14] have considered several methods to approximate a cyclic scheme by an α-acyclic one (by adding, deleting or modifying a minimum number of relation schemes) and have proved that some of these modifications imply NP-complete problems. A similar approach is given in [3], where the transformation is obtained by adding new attributes to the pre-existing ones.

Unfortunately, in several cases the database scheme obtained by means of such transformations does not represent in a natural way the realm of interest, which is intrinsically cyclic. On the other hand, in general, a user is not interested in the whole database scheme, but only in a proper subset of it, usually called a *view*. Therefore, it is meaningful to consider whether it is possible to find acyclic views in a cyclic scheme.

In this paper we deal with the following problem:

(i) Given a database scheme S and a subset A of its attributes, does an α-, γ- or Berge-acyclic view on S connecting A exist?

(ii) How is the complexity of the above problem influenced by the acyclicity degree of S?

A problem similar to the first one has been studied in [17] where a method to decompose a cyclic scheme in its α-acyclic and its cyclic part is given.

Related problems also arise in the implementation of a 'universal relation interface' [15, 19]. These interfaces provide a higher level of data independence to the users by allowing them to formulate queries in terms of attributes independently from the way in which attributes are clustered in the relation schemes.

In order to guarantee the correct and efficient execution of queries by a universal relation interface, we must consider the fact that a given set of attributes (which specify a query) may be connected in more than one way. Hence there may exist more than one interpretation of a query and more than one way to join database relations in order to answer the query. Both the unambiguity of a query and the existence of an efficient way to process the query are shown to be related to the acyclicity degree of the database scheme [1, 2, 6, 8, 12, 14].

In presence of cycles, the problem arises of imposing suitable assumptions either on the whole database or at least on a view on it. In particular, to provide a unique 'natural' interpretation of a query, Maier and Ullman [16] define suitable views (maximal objects) to break up cycles in the database scheme and give an algorithm which constructs such views starting from the dependencies that hold in the database.

In order to approach the solution of the problem stated at the beginning, we make use of the formalism of hypergraphs [4], which are often used for representing database schemes and studying their properties. After introducing the basic definitions we will prove that the problem of deciding the existence of acyclic views over general database schemes is NP-complete and that, even assuming that the scheme is α-acyclic, the problem of deciding the existence of a γ- or Berge-acyclic view remains computationally intractable. In the last part of this paper we show that if the database scheme is γ-acyclic efficient algorithms for deciding whether there exists a Berge-acyclic view connecting a given set of nodes may be found. Such algorithms are based on the concept of Bachman diagram [20] which, in the case of γ-acyclicity, provides a concise representation of the database scheme. In particular we show that the problem of the existence of a Berge-acyclic view, in this case, may be solved in $O(m+n)$ time, where m is the number of relation schemes and n the number of attributes appearing in the database scheme. It is worth noting that, by making use of the same algorithm, we may provide an efficient way of answering queries expressed in terms of attributes in a γ-acyclic scheme.

2. Basic definitions

We assume the reader to be familiar with basic concepts in relational database theory; the needed background is contained in [18]. In this section we introduce the formalism which will be used throughout the paper.

Definition 2.1. A *hypergraph* H is a pair $\langle N, E\rangle$, where N is a set of *nodes* and E a set of *edges* which are nonempty subsets of N, such that every node in N belongs to at least one edge in E. We say that a hypergraph $H'=\langle N', E'\rangle$ is a *subhypergraph* of H if E' is contained in E.

In a hypergraph the notions of connectivity and (connected) covering may be defined in a way similar to the case of graphs.

Definition 2.2. Let H be a hypergraph and let n, m be a pair of its nodes. We say that n and m are *connected* in H if they belong to the same edge or if there exists a node k in H connected to both n and m. A *hypergraph* is *connected* if all its nodes are pairwise connected (in this paper we consider only connected hypergraphs). A *covering* of a hypergraph $H=\langle N, E\rangle$ *over* a subset $\hat{N}$ of its nodes is a connected subhypergraph $H'=\langle N', E'\rangle$ of H such that $\hat{N}$ is contained in N'. We say that H' is a *nonredundant* covering over $\hat{N}$ if no proper subhypergraph of H' is still a covering over $\hat{N}$.

It is easy to see that the above concepts provide a natural representation of the relational theory concepts we are interested in. In fact, a database scheme can be represented by means of a hypergraph whose nodes and edges correspond to attributes and relation schemes respectively; furthermore, a view on a given set of attributes (i.e., a joinable set of relations that allow to answer a query involving a given set of attributes) can be represented by means of a covering over the set of nodes corresponding to these attributes.

As it was said in Section 1, while in the case of graphs the only kind of acyclic graphs are trees, in the case of hypergraphs several definitions of cycle and acyclic hypergraph have been introduced; in particular, in [8, 17] the following definitions are provided.

Definition 2.3. Let $H = \langle N, E \rangle$ be a hypergraph.
- A sequence $\langle e_1, \ldots, e_q \rangle$ of distinct edges in E is a *cycle* (or *Berge-cycle*) in H if $q > 1$ and $\Delta_i = e_i \cap e_{i+1} \neq \emptyset$, for $1 \leq i < q$, $\Delta_q = e_q \cap e_1 \neq \emptyset$, and $|\bigcup_{i=1}^{q} \Delta_i| \geq 2$.
- A cycle $\langle e_1, \ldots, e_q \rangle$ is a ***pure cycle*** if
  (i) $q = 3$ and $e_1 \cap e_2 \cap e_3 = \emptyset$, or
  (ii) $q > 3$ and for all pairs of indices i, j such that $1 < |i-j| < q-1$, we have $e_i \cap e_j = \emptyset$ (i.e., the edges which are nonadjacent in the sequence are disjoint).
- A *γ-cycle* is either a pure cycle or a cycle $\langle e_1, e_2, e_3 \rangle$ such that there exists a pair of nodes n, m in e_3 such that n is in e_1 and not in e_2 and m is in e_2 and not in e_1.
- A hypergraph is *ϑ-cyclic* if it has a ϑ-cycle; it is *ϑ-acyclic* otherwise (where ϑ = Berge or γ).
- A hypergraph is *α-acyclic* if it contains no cycle $\langle e_1, \ldots, e_q \rangle$ of length at least three such that for each triple of indices (i, j, k), $1 \leq i < j < k \leq q$, there is no edge in the hypergraph that contains $\Delta_i \cup \Delta_j \cup \Delta_k$; it is *cyclic* otherwise.

In [8] it is proved that Berge-acyclicity implies γ-acyclicity, γ-acyclicity implies α-acyclicity and that none of the reverse implications hold. In Figs. 1–4 cyclic and acyclic hypergraphs are shown.

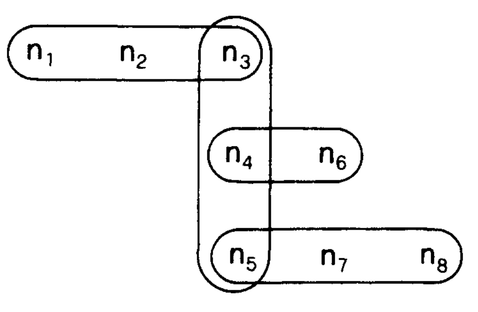


Fig. 1. A Berge-acyclic hypergraph.

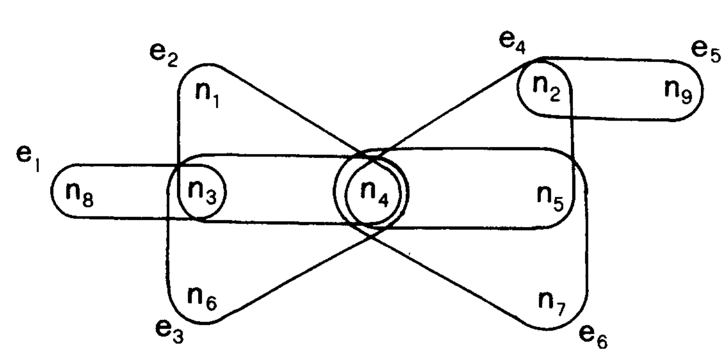


Fig. 2. A γ-acyclic, not Berge-acyclic hypergraph.

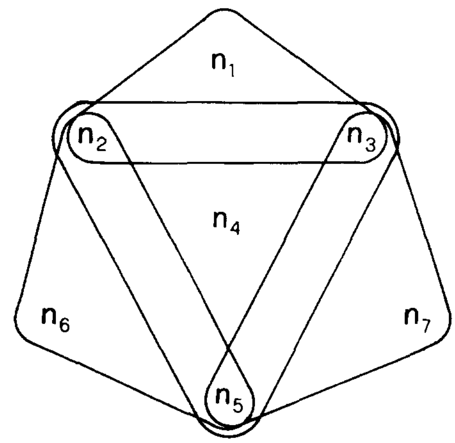


Fig. 3. An α-acyclic, γ-cyclic hypergraph.

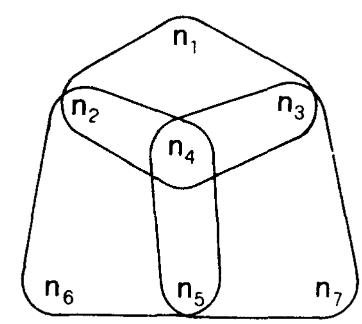


Fig. 4. A cyclic hypergraph.

A different equivalent definition of γ-acyclic hypergraph is based on the concept of Bachman diagram [20].

Definition 2.4. Let $H = \langle N, E\rangle$ be a hypergraph where $N = \{n_1, \ldots n_{|N|}\}$ is the set of nodes and $\{e_1, \ldots, e_{|E|}\}$ is the set of names of edges in E. We say that a directed graph $G = \langle V, A\rangle$ is a *Bachman diagram* of H, if:
- V contains the set $\{n_1, \ldots, n_{|N|}, e_1, \ldots, e_{|E|}\}$,
- there exists a bijection $f: P \to V$, where $P = \{\{n\} \mid n$ is in $N\} \cup \{X \mid X = \bigcap_{e \in E'} e$ for any subset E' of E and $X \neq \emptyset\}$, such that:
  - $f(\{n_i\}) = n_i$ for each n_i in N,
  - $f(X) = e_i$ if X is any edge in E with name e_i,
  - the arc $(f(X_i), f(X_j))$ is in A if and only if X_i is contained in X_j and for no X_k in P we have that X_k properly contains X_i and it is properly contained in X_j.

A hypergraph is γ-*acyclic* if it has a loop-free Bachman diagram [8].

Example 2.5. Let us consider the γ-acyclic hypergraph H given in Fig. 2. It is easy to see that if we consider the bijection given in Table 1, the directed graph G given in Fig. 5 provides a Bachman diagram of H.

Table 1
A bijection that provides a Bachman diagram for the hypergraph in Fig. 2.

P	$f(P)$
$\{n_1\}$	n_1
$\vdots$	$\vdots$
$\{n_9\}$	n_9
$\{n_3, n_8\}$	e_1
$\vdots$	$\vdots$
$\{n_4, n_5, n_7\}$	e_6
$\{n_3, n_4\}$	v_1
$\{n_4, n_5\}$	v_2

$$\begin{tikzcd}
 & & & n_1 \arrow[d] & & n_2 \arrow[d] \arrow[r] & e_5 & n_9 \arrow[l] \\
 & & & e_2 & & e_4 & & \\
n_8 \arrow[r] & e_1 & n_3 \arrow[l] \arrow[r] & v_1 \arrow[u] \arrow[d] & n_4 \arrow[l] \arrow[r] & v_2 \arrow[u] \arrow[d] & n_5 \arrow[l] & \\
 & & & e_3 & & e_6 & & \\
 & & & n_6 \arrow[u] & & n_7 \arrow[u] & &
\end{tikzcd}$$

Fig. 5. A Bachman diagram associated to the hypergraph in Fig. 2.

3. On the existence of acyclic coverings

In order to discuss the complexity of deciding whether a database scheme allows a ϑ-acyclic view ($\vartheta = \alpha$, γ, Berge), let us first prove the following lemma.

Lemma 3.1. *Let H be a general hypergraph and let $\hat{N}$ be a set of nodes. The problem of deciding whether there exists a covering over $\hat{N}$ without pure cycles is* NP-*complete.*

Proof. It is easy to see that the problem is in NP. The proof that it is also NP-hard may be obtained by reduction from the 3-SAT problem, that is the problem of deciding whether a propositional formula in conjunctive normal form with exactly three literals in every clause is satisfiable or not [11].

Let $w = (\boldsymbol{n}_{11} \vee \boldsymbol{n}_{12} \vee \boldsymbol{n}_{13}) \wedge \cdots \wedge (\boldsymbol{n}_{q1} \vee \boldsymbol{n}_{q2} \vee \boldsymbol{n}_{q3})$ be such a formula, where $\boldsymbol{n}_{ij}$ for $1 \leq i \leq q$ and $1 \leq j \leq 3$ belongs to the set of propositional variables or their negation. First of all let us consider a new formula w' which is obtained from w by introducing a dummy clause $\boldsymbol{n}_i = p$ between every two consecutive clauses c_i, c_{i+1}, where p is a new propositional variable. Clearly, w is satisfiable if and only if w' is. The reduction may be split in two steps. First of all we construct a graph $G = \langle V, A\rangle$ from w' in the following way:

$$V=\{n_{ij} \mid 1 \leqslant i \leqslant q, 1 \leqslant j \leqslant 3\} \cup \{n_i \mid 1 \leqslant i \leqslant q-1\} \cup \{n_0, n_q\},$$

$$\begin{aligned} A=&\{(n_{ij}, n_i) \mid 1 \leqslant i \leqslant q-1, 1 \leqslant j \leqslant 3\} \\ &\cup \{(n_i, n_{i+1\,j}) \mid 1 \leqslant i \leqslant q-1, 1 \leqslant j \leqslant 3\} \\ &\cup \{(n_0, n_{1j}) \mid 1 \leqslant j \leqslant 3\} \\ &\cup \{(n_q, n_{qj}) \mid 1 \leqslant j \leqslant 3\}. \end{aligned}$$

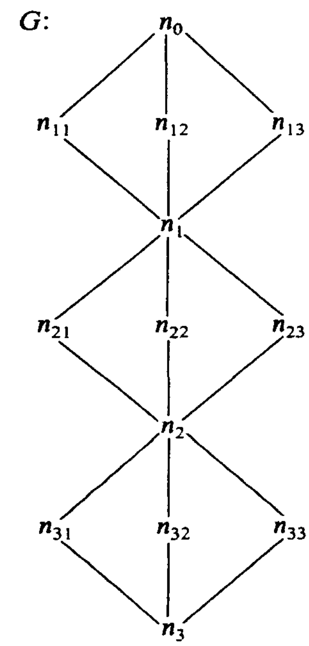
Besides we define a set of forbidden pairs of nodes in N,

$$F=\{(n_{ij}, n_{hk}) \mid n_{ij} \text{ is the negation of } n_{hk}\}.$$

Clearly, the problem of deciding whether w' is satisfiable or not is polynomially reducible to the problem of deciding whether there exists a path from n_0 to n_q in G which does not contain any forbidden pair of nodes (see Fig. 6).

$$w=(q_1 \vee q_2 \vee q_3) \wedge (\neg q_1 \vee q_2 \vee \neg q_3) \wedge (q_1 \vee \neg q_2 \vee \neg q_3)$$

$$w'=(q_1 \vee q_2 \vee q_3) \wedge p \wedge (\neg q_1 \vee q_2 \vee \neg q_3) \wedge p \wedge (q_1 \vee \neg q_2 \vee \neg q_3)$$



$$F=\{(n_{11}, n_{21}), (n_{12}, n_{32}), (n_{13}, n_{23}), (n_{13}, n_{33}), (n_{21}, n_{31}), (n_{22}, n_{32})\}$$

Fig. 6. A formula w and the corresponding graph. The path $n_0, n_{12}, n_1, n_{22}, n_2, n_{33}, n_3$ corresponds to the truth assignment $q_2=\mathrm{T}$, $q_3=\mathrm{F}$, $p=\mathrm{T}$ which satisfies w and w'.

Starting from the graph G we may now construct a hypergraph $H=\langle N, E\rangle$ in the following way:

$$\begin{aligned} N=&\{u_{ij}, v_{ij} \mid 1 \leqslant i \leqslant q, 1 \leqslant j \leqslant 3\} \cup \{u_0, u_q\} \cup \{w_i \mid 1 \leqslant i \leqslant q\} \\ &\cup \{f_{ijhk}=f_{hkij} \mid \text{for every } (n_{ij}, n_{hk}) \text{ in } F\} \\ &\cup \{u_i \mid 1 \leqslant i \leqslant q-1\}, \end{aligned}$$

$$E = \{e_0 = \{u_0, u_{11}, u_{12}, u_{13}\}, e_q = \{u_q, v_{q1}, v_{q2}, v_{q3}\}\}$$
$$\cup \{e_i = \{v_{i1}, v_{i2}, v_{i3}, u_{i+1\,1}, u_{i+1\,2}, u_{i+1\,3}, u_i\} \mid 1 \leq i \leq q-1\}$$
$$\cup \{e_{ij} = \{u_{ij}, v_{ij}, w_i\} \cup F_{ij} \mid 1 \leq i \leq q,\ 1 \leq j \leq 3,$$
$$\text{where } F_{ij} = \{f_{ijhk} \mid \text{for every } f_{ijhk} \text{ in } N\}\}.$$

Essentially, the hypergraph is dual with respect to the given graph since to every node in the graph we have a corresponding edge in the hypergraph, and two edges intersect if they correspond to adjacent nodes in G or if they correspond to forbidden pairs of nodes. Besides, all edges corresponding to nodes at the same level intersect in one node (see Fig. 7).

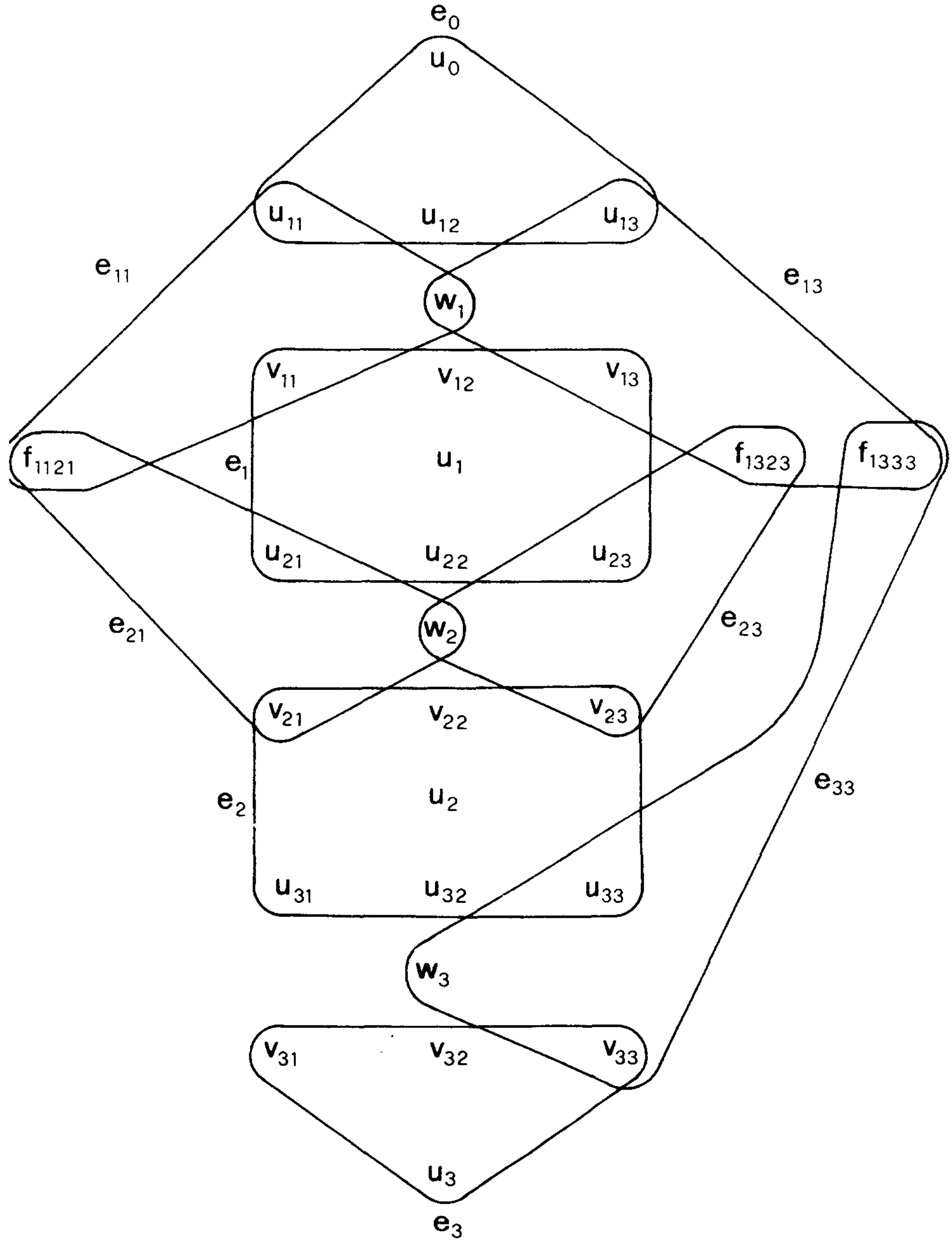


Fig. 7. The hypergraph corresponding to the formula w in Fig. 6 (note that for the sake of clarity only part of the edges are shown).

If we now consider the set of nodes

$$\hat{N} = \{u_i \mid 0 \leq i \leq q\} \cup \{w_i \mid 1 \leq i \leq q\},$$

we have that a covering of H over $\hat{N}$ exists without pure cycles if and only if there exists a path in G from n_0 to n_q which does not contain any forbidden pair of nodes. □

We can now state the first result concerning the existence of acyclic coverings over general hypergraphs.

Theorem 3.2. *The problem of deciding whether there exists a ϑ-acyclic covering ($\vartheta = \alpha$, γ, Berge) of a hypergraph H over a given set of nodes $\hat{N}$ is* NP*-complete.*

Proof. Since, by definition, γ- and Berge-acyclic hypergraphs do not contain pure cycles, we easily derive the correspondent NP-completeness result from Lemma 3.1. Furthermore, let us consider a hypergraph H as in the proof of Lemma 3.1. We prove that a covering of H is α-acyclic if and only if it does not contain pure cycles. In fact, by definition, a hypergraph is cyclic if it contains a cycle $\langle e_1, \ldots, e_q \rangle$ of length at least three such that for each triple of indices (i, j, h), $1 \leq i < j < h \leq q$, there is no edge in the hypergraph that contains $\Delta_i \cup \Delta_j \cup \Delta_h$ (where $\Delta_k = e_k \cap e_{k+1}$ if $1 \leq k < q$ and $\Delta_q = e_1 \cap e_q$).

If an α-acyclic hypergraph contains a pure cycle $\langle e_1, \ldots, e_q \rangle$, then every e_i, $1 \leq i \leq q$, must contain at least a pair of distinct nodes that belong to three or more edges.

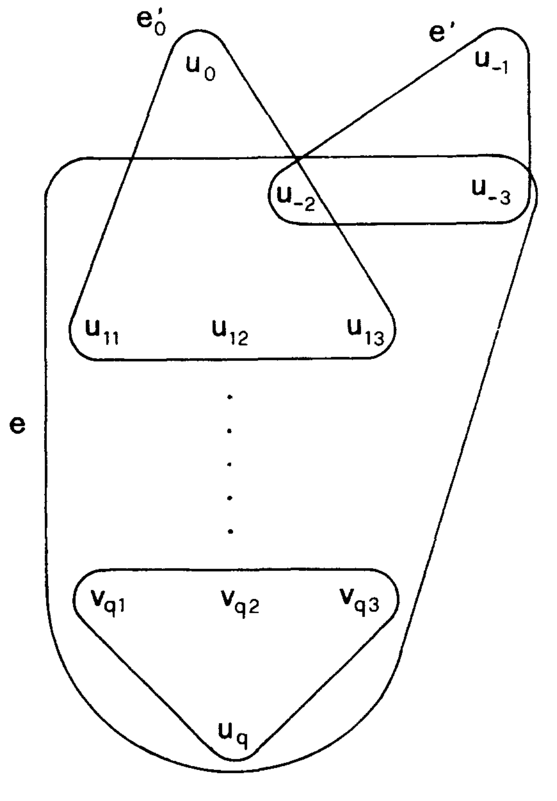


Fig. 8. The α-acyclic hypergraph obtained by modifying the hypergraph in Fig. 7.

Since all edges in H contain at most one node with the above property, the theorem is proved. □

Let us now consider the case in which the given hypergraph is α-acyclic. We will show that the problem of the existence of a ϑ-acyclic covering ($\vartheta=\gamma$, Berge) over a given set of nodes remains computationally intractable also in this case.

Theorem 3.3. *Let H be an α-acyclic hypergraph and $\hat{N}$ a subset of its nodes. The problem of determining whether H has a ϑ-acyclic covering ($\vartheta=\gamma$, Berge) over $\hat{N}$ is* NP-*complete.*

Proof. We derive the result by modifying the proof of Lemma 3.1. In fact, let us consider the hypergraph $H=\langle N, E\rangle$ and the set of nodes $\hat{N}$ in the proof of Lemma 3.1 and let $H'=\langle N', E'\rangle$ and $\hat{N}'$ be obtained from H and $\hat{N}$ in the following way (see Fig. 8):

$$N'=N\cup\{u_{-1}, u_{-2}, u_{-3}\},$$

$$E'=(E-\{e_0\})\cup\{e_0'=e_0\cup\{u_{-2}\}, e=N'-\{u_0, u_{-1}\}, e'=\{u_{-1}, u_{-2}, u_{-3}\}\},$$

$$\hat{N}'=\hat{N}\cup\{u_{-1}\}.$$

It is easy to see that, because of the existence of edge e in E', H' is α-acyclic. Since every covering of H' over $\hat{N}'$ must contain e_0' and e', neither a γ-acyclic covering nor a Berge-acyclic covering can contain e (in fact, $\langle e_0', e', e\rangle$ is a γ-cycle and $\langle e_0', e\rangle$ is a Berge-cycle). Therefore, if it was possible to solve our problem in polynomial time, it would be possible to decide the problem in Lemma 3.1 in polynomial time. In fact, a covering of H over $\hat{N}$ without pure cycles may be obtained by eliminating the edges e_0', e' from the set of edges of a γ-acyclic covering of H' over $\hat{N}'$ and by adding the edge e_0. □

The negative results seen insofar are intrinsically related to the structure of the considered hypergraphs. In fact, if we consider the classes of hypergraphs satisfying stronger acyclicity conditions, the problem becomes more easily solvable.

Theorem 3.4. *The problem of deciding whether a γ-acyclic hypergraph $H=\langle N, E\rangle$ allows a Berge-acyclic covering over a given set of nodes $\hat{N}$ is solvable in* $O(|E|^2|N|)$ *time.*

Proof. In [1] it was proved that if $H=\langle N, E\rangle$ is a γ-acyclic hypergraph and $H'=\langle N', E'\rangle$, $H''=\langle N'', E''\rangle$ are two nonredundant coverings of H over $\hat{N}$, for every edge e' in E' there exists an edge e'' in E'' such that e'' has the same intersections as e' with the remaining edges in E' and vice versa (i.e., two nonredundant coverings differ only for nodes belonging only to one edge in the covering).

In order to decide whether there exists a Berge-acyclic covering of H over $\hat{N}$ we may use the following algorithm:

Step 1. Determine a nonredundant covering $\hat{H}$ of H over $\hat{N}$ by repeating the following step:

1.1. delete (in any order) an edge e if it satisfies the following redundancy predicate:

"$e \cap \hat{N} \subseteq \bigcup_{\hat{e} \in E-\{e\}} \hat{e}$ and the nodes in $\hat{N}$ are pairwise connected after the deletion of e".

until no more edges e satisfy the redundancy predicate.

Step 2. Test whether $\hat{H}$ is Berge-acyclic or not.

In fact, from the above stated property of γ-acyclic hypergraphs it follows that either all nonredundant coverings of H over $\hat{N}$ are Berge-acyclic or no covering of H over $\hat{N}$ is Berge-acyclic.

Since the redundancy predicate may be tested in $O(|E||N|)$ time, Step 1 takes $O(|E|^2|N|)$ time. It is easy to see that Step 2 requires $O(|E|+|N|)$ time. The overall complexity is then $O(|E|^2|N|)$ time. □

It is interesting to observe that the above problem may be approached in a more efficient way if we assume that the given hypergraph is represented by means of a Bachman diagram. Note that a Bachman diagram $G = \langle V, A\rangle$ provides a representation of a hypergraph $H = \langle N, E\rangle$ by means of the following rules:

- N is the set of source vertices in V;
- E is the set of edges obtained by considering for each vertex e_i in V the set of source vertices n_j such that there exists a directed path from n_j to the vertex e_i. No other edge is in E.

The advantage of using such a representation arises from the fact (proved in [20]) that the Bachman diagram of a γ-acyclic hypergraph has a size bounded by $O(|E|+|N|)$ and, hence, provides a much shorter representation of a given hypergraph, while in general a hypergraph requires a representation of size $O(|E||N|)$. Note that such a property of γ-acyclic hypergraphs does not hold for α-acyclic hypergraphs for which the Bachman diagram may have an exponential size in the size of H.

Theorem 3.5. *Given a Bachman diagram of a γ-acyclic hypergraph $H = \langle N, E\rangle$, the problem of deciding whether H allows a Berge-acyclic covering over a given set of nodes may be solved in* $O(|E|+|N|)$ *time.*

Proof. Let $G = \langle V, A\rangle$ be a Bachman diagram of H. Let us first determine a minimum connected subgraph $\hat{G}$ of G containing $\hat{N}$. Since the graph G is loop-free, $\hat{G}$ is also loop-free and may be obtained by means of a backtracking algorithm over G in $O(|E|+|N|)$ time. Let us now construct a nonredundant covering $\hat{H}$ of H over

$\hat{N}$ in the following way:

- consider each sink s in $\hat{G}$;
  - if s is also a sink in G, then s corresponds to an edge in H and such an edge must belong to any covering of H over $\hat{N}$ (otherwise, the nodes in $\hat{N}$ would not be connected);
  - otherwise, s corresponds to a set of nodes of H which belong to all coverings of H over $\hat{N}$; due to the properties of γ-acyclic hypergraphs it is hence necessary to provide only one edge in E containing such nodes; this may be accomplished by following any directed path from s to a sink in G.

Furthermore, if there exists a Berge-acyclic covering of H over $\hat{N}$ the following property must hold:

- the only case in which two or more arcs in $\hat{G}$ leave a vertex v is when v is a source vertex in G;

because otherwise such a vertex v would correspond to an intersection in $\hat{H}$ of two or more edges containing two or more nodes.

Since both steps above just require the visit of the Bachman diagram, the theorem is proved. □

Let us consider, for example, the hypergraph of Fig. 2. If we wish to provide a covering over n_5, n_8, and n_9, by considering the relative Bachman diagram (Fig. 5), we may determine the minimum subgraph $\hat{G}$ spanning from n_5 to n_8 and n_9, in which e_1, e_4, and e_5 are sink nodes in both G and $\hat{G}$, and v_1 is a sink node only in $\hat{G}$. Starting from v_1 we provide a completion of a nonredundant covering $\hat{H}$ by adding edge e_2. Since the property stated in the proof of the theorem holds, we may conclude that the edges e_1, e_2, e_4, and e_5 provide a Berge-acyclic covering over n_5, n_8, and n_9.

4. Conclusions

In various applications of relational databases it is relevant to determine views on the database scheme which satisfy particular acyclicity conditions. In this paper, by representing database schemes in terms of hypergraphs, it has been shown that this problem is computationally hard. In fact, in Theorem 3.2 we proved that the problem of deciding the existence of a ϑ-acyclic view ($\vartheta = \alpha$, γ, Berge) over a given set of attributes is NP-complete, and in Theorem 3.3 we showed that even if the database scheme itself is α-acyclic, the problem is still NP-complete.

These results show that the problems of the unambiguous and efficient processing of queries which are related to the existence of acyclic views in the database scheme may be easily solved only if stronger conditions than α-acyclicity are imposed on the scheme. In fact, Theorems 3.4 and 3.5 show polynomial algorithms for deciding whether a γ-acyclic scheme admits a Berge-acyclic view.

Note that the procedure described in the proof of Theorem 3.5 provides a constructive approach to the determination of a nonredundant covering of a γ-acyclic hypergraph. Since in a γ-acyclic hypergraph all nonredundant coverings consist in the same number of edges [1], the same procedure allows us to determine a covering of a γ-acyclic consisting of the minimum number of edges, in $O(|E|+|N|)$ time. This result is particularly relevant for providing an efficient way of minimizing the number of join operations required to answer a query in a γ-acyclic database.

References

[1] G. Ausiello, A. d'Atri and M. Moscarini, Minimal coverings of acyclic database schemata, in: H. Gallaire et al., eds., *Advances in Database Theory, Vol. 2* (Plenum Press, New York, 1984) 27-52.

[2] C. Beeri, R. Fagin, D. Maier and M. Yannakakis, On the desirability of acyclic database schemes, *J. ACM* **30** (3) (1983) 479-573.

[3] C. Beeri and M. Kifer, Elimination of intersection anomalies from database schemas, *Proc. 2nd ACM SIGACT-SIGMOD Symp. on Principles of Database Systems,* Atlanta, GA (1983) 340-351.

[4] C. Berge, *Graphs and Hypergraphs* (North-Holland, Amsterdam, 1973).

[5] A. d'Atri and M. Moscarini, On the recognition and design of acyclic databases, *Proc. 3rd ACM SIGACT-SIGMOD Conf. on Principles of Database Systems,* Waterloo, Canada (1984) 1-8.

[6] A. d'Atri, M. Moscarini and N. Spyratos, Answering queries in relational databases, *Proc. Annual Meeting of SIGMOD: SIGMOD Record* **13** (4) (1983) 173-177.

[7] R. Fagin, Acyclic database schemes (of various degree): A painless introduction, *Proc. 8th Colloquium on Trees in Algebra and Programming,* Lecture Notes in Computer Science **159** (Springer, Berlin, 1983) 65-89.

[8] R. Fagin, Degree of acyclicity for hypergraphs and relational database schemes, *J. ACM* **30** (3) (1983) 514-550.

[9] R. Fagin, A. O. Mendelzon and J. D. Ullman, A simplified universal relation assumption and its properties, *ACM Trans. Database Systems* **7** (3) (1982) 343-360.

[10] H. Gallaire, J. Minker and J.M. Nicolas, eds., *Advances in Database Theory, Vol. 2* (Plenum Press, New York, 1984).

[11] M.R. Garey and D. Johnson, *Computers and Intractability* (Freeman, San Francisco, CA, 1979).

[12] N. Goodman and O. Shmueli, Tree queries: A simple class of queries, *ACM Trans. Database Systems* **7** (4) (1982) 653-657.

[13] N. Goodman and O. Shmueli, Transforming cyclic schemas into trees, *Proc. 1st ACM SIGACT-SIGMOD Symp. on Principles of Database Systems,* Los Angeles, CA (1982) 49-54.

[14] N. Goodman, O. Shmueli and Y.T. Tai, GYO reductions, connections, tree and cyclic schemas and tree projections, *Proc. 2nd ACM SIGACT-SIGMOD Symp. on Principles of Database Systems,* Atlanta, GA (1983) 267-278.

[15] D. Maier, D. Rozenshtein and D.S. Warren, Windows on the world, *Proc. Annual Meeting of SIGMOD; SIGMOD Record* **13** (4) (1983) 68-78.

[16] D. Maier and J.D. Ullman, Maximal objects and the semantics of universal relation databases, *ACM Trans. Database Systems* **8** (1) (1983) 1-14.

[17] J. Paradaens and D. Van Gucht, An application of the theory of graphs and hypergraphs to the decomposition of relational database schemas, *Proc. 8th Colloquium on Trees in Algebra and Programming,* Lecture Notes in Computer Science **159** (Springer, Berlin, 1983) 65-89.

[18] J.D. Ullman, *Principles of Database Systems* (Computer Science Press, Potomac, MD, 2nd ed., 1982).

[19] J.D. Ullman, Universal relation interfaces for database systems, *IFIP Conf.,* Paris, France (1983) 243-252.

[20] M. Yannakakis, Algorithms for acyclic database schemes, *Proc. 7th Conf. on Very Large Data Bases,* Cannes, France (1981) 82-94.